# Multi-sensor PHD filtering with application to sensor management

Emmanuel Delande

N° d'ordre : 172

Ecole Centrale de Lille

# THÈSE

présentée en vue
d'obtenir le grade de

# DOCTEUR

Spécialité : Automatique, Génie Informatique,
Traitement du Signal et Image

par

**Emmanuel Delande**

Doctorat délivré par l'Ecole Centrale de Lille

---

# Filtrage PHD multicapteur avec application à la gestion de capteurs

---

Soutenue le 30 janvier 2012 devant le jury d'examen :

| | | | |
|---|---|---|---|
| **Président** | M. Pierre Del Moral | Directeur de recherche | INRIA |
| **Rapporteur** | M. Daniel E. Clark | Lecturer | Heriot-Watt University |
| **Rapporteur** | M. Simon J. Godsill | Professor | University of Cambridge |
| **Membre** | M. Frédéric Dambreville | Docteur HDR | DGA |
| **Invité** | M. Dominique Heurguier | Ingénieur de recherche | Thales (TCS) |
| **Directeur de thèse** | M. Emmanuel Duflos | Professeur | Ecole Centrale de Lille |
| **Co-directeur de thèse** | M. Philippe Vanheeghe | Professeur | Ecole Centrale de Lille |

Préparée au Laboratoire d'automatique, génie informatique et signal
LAGIS UMR CNRS 8219

Ecole Doctorale SPI 072 (Lille I, Lille III, Artois, ULCO, UVHC, EC Lille)
**PRES Université Lille Nord-de-France**

**Filtrage PHD multicapteur avec application à la gestion de capteur**

**Résumé**
Le filtrage multiobjet est une technique de résolution du problème de détection et/ou suivi dans un contexte multicible. Cette thèse s'intéresse au filtre PHD (Probability Hypothesis Density), une célèbre approximation du filtre RFS (Random Finite Set) adaptée au cas où les observations sont le fruit d'un seul capteur. La première partie propose une construction rigoureuse du filtre PHD multicapteur exact et son expression simplifiée, sans approximation, grâce à un partitionnement joint de l'espace d'état des cibles et des capteurs. Avec cette nouvelle méthode, la solution exacte du filtre PHD multicapteur peut être propagée dans des scénarios de surveillance simples. La deuxième partie aborde le problème de gestion des capteurs dans le cadre du PHD. A chaque itération, le BET (Balanced Explorer and Tracker) construit une prédiction du PHD multicapteur a posteriori grâce au PIMS (Predicted Ideal Measurement Set) et définit un contrôle multicapteur en respectant quelques critères opérationnels simples adaptés aux missions de surveillance.

**Mots-clés : filtrage multiobjet, PHD multicapteur, gestion de capteurs**

---

**Multi-sensor PHD filtering with application to sensor management**

**Abstract**
The aim of multi-object filtering is to address the multiple target detection and/or tracking problem. This thesis focuses on the Probability Hypothesis Density (PHD) filter, a well-known tractable approximation of the Random Finite Set (RFS) filter when the observation process is realized by a single sensor. The first part proposes the rigorous construction of the exact multi-sensor PHD filter and its simplified expression, without approximation, through a joint partitioning of the target state space and the sensors. With this new method, the exact multi-sensor PHD can be propagated in simple surveillance scenarii. The second part deals with the sensor management problem in the PHD framework. At each iteration, the Balanced Explorer and Tracker (BET) builds a prediction of the posterior multi-sensor PHD thanks to the Predicted Ideal Measurement Set (PIMS) and produces a multi-sensor control according to a few simple operational principles adapted to surveillance activities.

**Keywords: multi-object filtering, multi-sensor PHD, sensor management**

# Acknowledgements

My first thanks go to my supervisors, Prof. Emmanuel Duflos and Prof. Philippe Vanheeghe, and my closest collaborator in Thales, Dominique Heurguier. Each in their own way, they provided me with invaluable guidance and knowledge throughout these three years. Given my indefectible lack of self-confidence, I can only imagine how demanding this must have been and I am much indebted to them for their patience and their continuous support.

I am also very grateful to Prof. Simon Godsill and Dr. Daniel Clark for their in-depth review of my manuscript; their numerous suggestions and constructive advice were much useful and helped me shape this dissertation in its final form. Prof. Pierre Del Moral and Dr. Frédéric Dambreville, who took interest in my work as members of the defense committee, have also my deepest gratitude.

Even though I cannot mention every one of them personally, I gratefully thank all the members of both the LAGIS in Ecole Centrale de Lille and the SEA in Thales Communications & Security who were involved in my research activities one way or another. Through their scientific expertise, their technical insight, or simply their joyful mood, they all contributed to this eventful experience I had the priviledge of being a part of. My special thanks to Christine Yvoz and the late Marie-Françoise Tricot for their tremendous support in the organization of my work between the LAGIS and the SEA.

Finally, a very special thanks go to my parents, my family, and my friends, who helped me get through tough times and without whom this work would not have been possible.

# Résumé

Les quatres sections de ce résumé suivent les quatres chapitres de la thèse. La première section reprend les grandes lignes de la théorie des ensembles aléatoires finis et décrit les équations du filtre PHD monocapteur. La deuxième section présente une contribution clé de la thèse, la construction du filtre PHD multicapteur et la simplification de son écriture sans approximation grâce à une méthode de partitionnement joint de l'espace d'état des cibles et de l'ensemble des capteurs. La troisième section résume la construction d'un gestionnaire de capteurs dans le cadre du PHD, le BET (Balanced Explorer and Tracker). Enfin, la quatrième section expose brièvement l'implémentation particulaire du filtre PHD multicapteur et du contrôleur BET, ainsi que les principaux résulats de simulation.

## La théorie des ensembles aléatoires finis

Dans le contexte de la détection et du suivi de cibles, la théorie des *ensembles aléatoires finis* (Random Finite Sets) permet une représentation originale de la situation opérationelle ; autrement dit, du nombre et de l'état des différentes cibles évoluant dans la zone de surveillance. Contrairement aux techniques classiques qui assignent une *piste* pour chaque cible détectée et la maintiennent à jour avec les mesures successives, la théorie RFS décrit l'*ensemble* des cibles à un instant donné comme la réalisation d'un ensemble aléatoire fini, c'est à dire une *unique* variable aléatoire dont le nombre d'éléments - le nombre de cibles - et la valeur des éléments - l'état des cibles - sont aléatoires. Plus précisément, on définit un RFS comme une fonction mesurable [Vo 08]:

$$\begin{aligned} \Xi \ : \ & \Omega \to \mathcal{F}(\mathcal{X}) \\ & \omega \mapsto X = \Xi(\omega) \end{aligned} \tag{1}$$

où $(\Omega, \sigma(\Omega), \mathbb{P})$ est un espace probabilisé équipé de la topologie de Matheron, et $\mathcal{F}(\mathcal{X})$ l'espace des sous-ensembles finis de l'espace d'état monocible $\mathcal{X}$.

Le filtrage RFS consiste à propager au cours du temps des RFSs décrivant la configuration multicible (nombre de cibles *et* état de ces dernières) en fonction de l'arrivée des nouvelles mesures. Au prix d'hypothèses relativement faibles sur la modélisation des cibles et des capteurs, le filtre RFS permet en théorie :

- de rassembler dans un seul objet aléatoire les méchanismes de naissance, évolution et disparition des cibles ;
- à chaque itération, de faire évoluer cet objet aléatoire selon le schéma bayésien classique "prédiction $\rightarrow$ mise à jour".

Quatre RFSs interviennent à chaque itération dans les équations bayésiennes :

1. Le *RFS (multicible) de transition* $\Xi^T_{k,k+1}(X)$, avec loi de probabilité $p_{\Xi^T_{k,k+1}}(.|X)$, décrit la configuration multicible à l'instant $k+1$ conditionnellement à un ensemble multicible $X$ à l'instant $k$.

2. Le *RFS (multicible) a priori* $\Xi_{k+1|k}$, avec loi de probabilité $p_{\Xi_{k+1|k}}(.|Z_{1:k})$, décrit la configuration multicible à l'instant $k+1$ conditionnellement aux mesures produites jusqu'à l'instant $k$.

3. Le *RFS (multimesure) d'observation* $\Sigma_{k+1}(X)$, avec loi de probabilité $p_{\Sigma_{k+1}}(.|X)$, décrit la configuration multimesure à l'instant $k+1$ conditionnellement à un ensemble multicible $X$ à l'instant $k+1$.

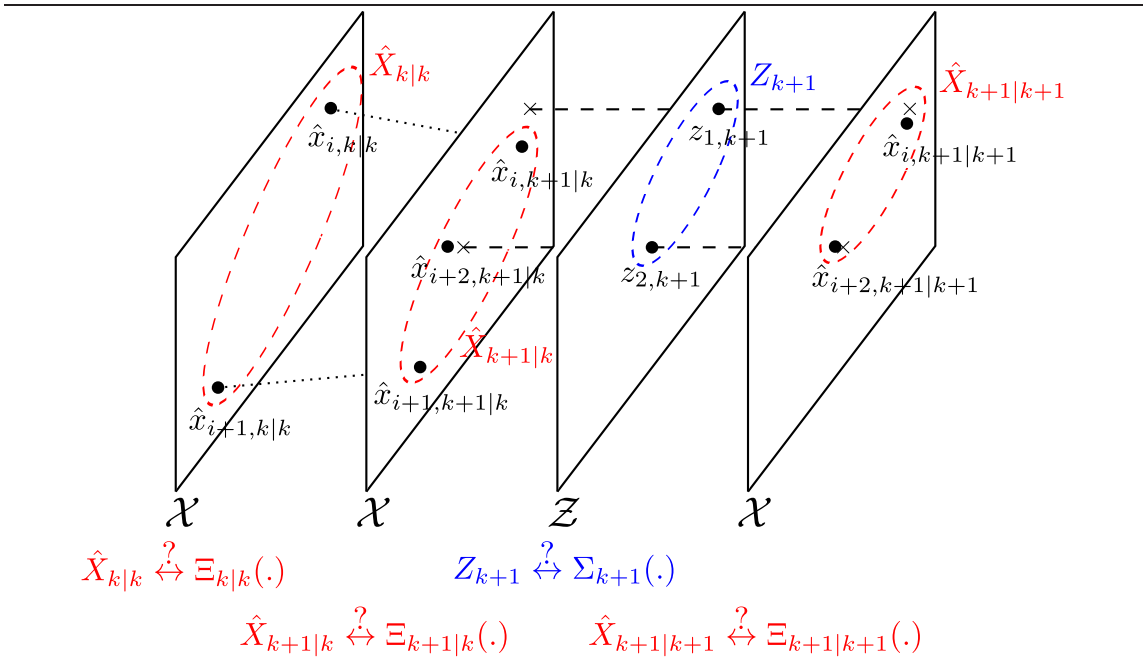
4. Le *RFS (multicible) a posteriori* $\Xi_{k+1|k+1}$, avec loi de probabilité $p_{\Xi_{k+1|k+1}}(.|Z_{1:k+1})$, décrit la configuration multicible à l'instant $k+1$ conditionnellement aux mesures produites jusqu'à l'instant $k+1$.

Ces quatres RFSs sont liés par les équations du filtre RFS [Mahl 02]:

$$p_{\Xi_{k+1|k}}(.|Z_{1:k}) = \int_{\mathcal{F}(\mathcal{X})} p_{\Xi^T_{k,k+1}}(.|X) p_{\Xi_{k|k}}(X|Z_{1:k}) \mu(dX) \tag{2}$$

$$p_{\Xi_{k+1|k+1}}(.|Z_{1:k+1}) = \frac{p_{\Sigma_{k+1}}(Z_{k+1}|.) p_{\Xi_{k+1|k}}(.|Z_{1:k})}{\int_{\mathcal{F}(\mathcal{X})} p_{\Sigma_{k+1}}(Z_{k+1}|X) p_{\Xi_{k+1|k}}(X|Z_{1:k}) \mu(dX)} \tag{3}$$

Le principal avantage de cette méthode devant les techniques usuelles est qu'elle ne nécessite ni heuristiques pour la création et la destruction de pistes ni association explicite entre mesures et pistes. En d'autres termes, il suffit d'implémenter les équations bayésiennes de *prédiction* (2) et *mise à jour* (3) pour propager les RFSs décrivant la configuration multicible.




On peut remarquer que la structure du filtre RFS est similaire à celle du filtre monocible classique, les fonctions monoobjet (fonction de transition, de vraisemblance, etc.) étant remplacées par leur "équivalent" multiobjet. Mais, si les fonctions monoobjet peuvent être construites explicitement au prix d'hypothèses "raisonnables" sur le comportement des cibles et le fonctionnement interne des capteurs, leur équivalent multiobjet sont en règle générale inexploitables. Par exemple, étant donné que le RFS de transition $\Xi^T_{k,k+1}$ comprend le méchanisme de naissance, de disparition et d'évolution des cibles, la quantité $p_{\Xi^T_{k,k+1}}(X|Y)$ doit être déterminée pour des ensembles $X$ et $Y$ quelconques, et donc en particulier de taille quelconque. Indépendamment de la complexité des lois de probabilités, les intégrales sur $\mathcal{F}(\mathcal{X})$, évidemment plus complexes à traiter que les intégrales sur $\mathcal{X}$ présentes dans les équations bayésiennes monocible, sont impraticables sauf cas bien particuliers - par exemple, si le nombre de cibles est fixe.

Le filtre PHD est une approximation du filtre RFS qui restreint la propagation d'information sur le RFS a priori $\Xi_{k+1|k}$ (resp. a posteriori $\Xi_{k+1|k+1}$) à son premier moment ou PHD $v_{\Xi_{k+1|k}}(.|Z_{1:k})$ (resp. $v_{\Xi_{k+1|k+1}}(.|Z_{1:k+1})$) plutôt qu'à sa loi de probabilité $p_{\Xi_{k+1|k}}(.|Z_{1:k})$ (resp. $p_{\Xi_{k+1|k+1}}(.|Z_{1:k+1})$). Les équations du filtre PHD, *dans le*

*cas monocapteur uniquement*, sont les suivantes [Mahl 03a] :

$$v_{\Xi_{k+1|k}}(.|Z_{1:k}) = \int_{\mathcal{X}} \left(p^s_{k,k+1}(x) f^t_{k,k+1}(.|x) + \lambda^s_{k,k+1}(x) s_{k,k+1}(.|x)\right) v_{\Xi_{k|k}}(x|Z_{1:k})dx + \lambda^b_{k,k+1} b_{k,k+1}(.) \tag{4}$$

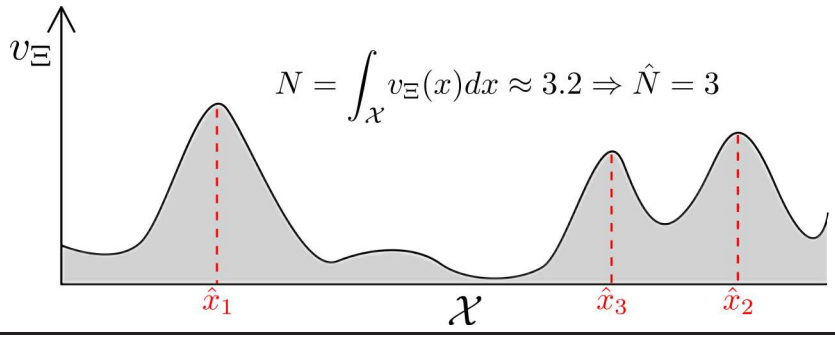
$$v_{\Xi_{k+1|k+1}}(.|Z_{1:k+1}) = \left(1 - p^d_{k+1}(.) + \sum_{z \in Z_{k+1}} \frac{p^d_{k+1}(.) L^z_{k+1}(.)}{\lambda^c_{k+1} c_{k+1}(z) + v_{\Xi_{k+1|k}}[p^d_{k+1} L^z_{k+1}]}\right) v_{\Xi_{k+1|k}}(.|Z_{1:k}) \tag{5}$$

avec $v_{\Xi_{k+1|k}}[p^d_{k+1} L^z_{k+1}] \stackrel{def}{=} \int_{\mathcal{X}} p^d_{k+1}(x) L^z_{k+1}(x) v_{\Xi_{k+1|k}}(x|Z_{1:k})dx$. Comparées à leur équivalent RFS (2), (3), les équations (4), (5) sont plus simples à manipuler car les fonctions multiobjet ont disparu au profit de fonctions monoobjet plus classiques. Par exemple, la loi de probabilité du RFS de transition $p_{\Xi^T_{k,k+1}}$ est remplacée par une fonction de transition $f^t_{k,k+1}$, une probabilité de survie $p^s_{k,k+1}$, une intensité de spwaning $\lambda^s_{k,k+1} s_{k,k+1}$ et de naissance spontanée $\lambda^b_{k,k+1} b_{k,k+1}$, toutes monocible. De cette façon, le filtre PHD peut être implémenté avec des méthodes d'approximation similaires à celles employées dans le filtre monocible (filtrage particulaire notamment, voir plus loin).

Le passage du filtre RFS à l'approximation que constitue le filtre PHD nécessite plusieurs hypothèses ; certaines sont usuelles dans les problèmes de détection - indépendance des cibles, indépendance des mesures, etc. - d'autres plus spécifiques au cadre du RFS. En particulier, l'hypothèse de Poisson suppose que les RFSs multicible $\Xi_{k+1|k}$ et $\Xi_{k+1|k+1}$ appartiennent à une classe restreinte de RFS *complètement décrits* par leur PHD, les Poisson RFSs. C'est à dire, connaissant le PHD $v_{\Xi_{k+1|k}}(.|Z_{1:k})$, le RFS $\Xi_{k+1|k}$ est entièrement décrit par :

- la loi décrivant le nombre de cibles, Poisson de paramètre $v_{\Xi_{k+1|k}}[1]$ ;
- la loi décrivant la distribution des cibles dans $\mathcal{X}$, chaque cible étant indépendamment et identiquement distribuée selon la densité $\frac{v_{\Xi_{k+1|k}}(.|Z_{1:k})}{v_{\Xi_{k+1|k}}[1]}$.

Dans la grande majorité des cas d'utilisation d'un filtre multicible, la donnée de sortie du filtre doit être un ensemble de cibles qu'il faut donc extraire du PHD. Grâce aux propriétés ci-dessus, l'estimation du nombre de cible est donnée par l'intégrale du PHD sur l'ensemble de la zone - $v_{\Xi_{k+1|k}}[1]$ - et les cibles peuvent être placées autour des extrema locaux comme illustré ci-après :




A noter que d'autres processus d'extraction sont possibles et certains sont décrits plus en détail dans la thèse. De façon générale, l'extraction est *indépendante* et n'est pas nécessaire à la propagation du PHD ; elle ne fait donc pas partie intégrante du filtre PHD.

## Filtrage PHD multicapteur

L'extension du filtre PHD au cas multicapteur est une contribution importante de cette thèse. En reprenant le principe de la construction du filtre monocapteur et en supposant que le processus d'observation de chaque capteur est indépendant conditionnellement à la configuration des cibles, on peut obtenir l'expression *exacte* de l'équation de mise à jour :

$$v_{\Xi_{k+1|k+1}}(x|Z_{1:k+1}) = \beta[\delta_\emptyset, \delta_x]K_{\mathcal{X}}^{-1} + \frac{\sum_{C\in\mathcal{C}(Z_{k+1})}\sum_{C_i\in C}\left(\beta[\delta_{C_i},\delta_x]\prod_{C_j\neq C_j}\beta[\delta_{C_j},1]\right)}{\sum_{C\in\mathcal{C}(Z_{k+1})}\prod_{C_i\in C}\beta[\delta_{C_i},1]}K_{\mathcal{X}}^{-1} \tag{6}$$

où :

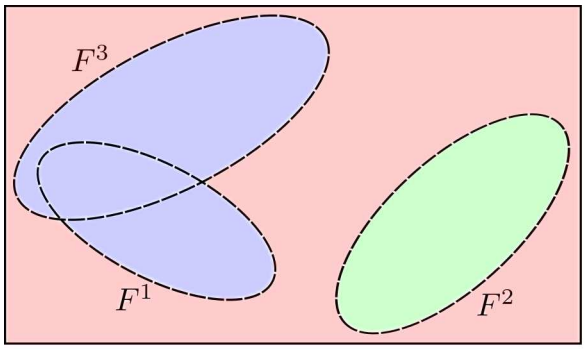
- $Z_{k+1} = \bigsqcup_{j=1}^{S} Z_{k+1}^j$ est l'ensemble de mesures des $S$ capteurs;
- $\mathcal{C}(Z_{k+1})$ est l'ensemble des *termes combinatoires* construit sur l'ensemble des mesures.

L'association implicite entre mesures et cibles est illustrée par la présence des termes combinatoires, représentant l'ensemble des regroupements possibles entre mesures de différents capteurs mais originaires de la même cible. Les *cross-terms* $\beta$ sont des fonctionnelles, chacune pondérant l'association entre un regroupement de mesures et un point de l'espace d'état *conditionnellement au PHD a priori* $v_{\Xi_{k+1|k}}(.|Z_{1:k})$. Par exemple :

- $\beta[\delta_{\emptyset}, \delta_x]$ : une cible existe en $x$, et n'est pas détectée ;
- $\beta[\delta_{\{z_1^1, z_2^3\}}, 1]$ : les mesures $z_1^1$ et $z_2^3$ proviennent de la même cible, dont l'état est inconnu ;
- $\beta[\delta_{\{z_1^1, z_2^3\}}, \delta_x]$ : une cible existe en $x$, est à l'origine de la mesure $z_1^1$ du capteur 1, à l'origine de la mesure $z_2^3$ du capteur 3, n'est pas détectée par les autres capteurs.

A noter que le PHD a priori n'apparait pas explicitement dans l'équation de mise à jour multicapteur mais est utilisé dans la construction des cross-terms. L'équation de mise à jour multicapteur est intéressante sur le plan théorique car elle donne l'expression exacte du PHD a posteriori ; en substituant (6) à (5), on construit un filtre propageant le PHD multicapteur *sans approximation*, une référence précieuse pour comparer et étudier les approximations multicapteur usuelles (voir plus loin).

Sur le plan pratique, le coût algorithmique de la construction de l'ensemble des termes combinatoires explose avec l'augmentation du nombre de mesures et/ou de capteurs, et le filtre PHD multicapteur exact n'est pas directement exploitable dans un algorithme de poursuite en temps réel. Toutefois, l'expression de la mise à jour peut être simplifiée, sans approximation, en considérant un partitionnement joint de l'espace d'état et de l'ensemble des capteurs reposant sur la configuration des champs de vue des capteurs.




$P = 2$ :

$T(0) = \mathcal{X} \setminus \bigcup_{j=1}^{3} F^j$

$S(1) = \{2\}$ $\quad T(1) = F^2$

$S(2) = \{1, 3\}$ $\quad T(2) = F^1 \cup F^3$

Dans la figure ci-dessus, par exemple, l'équation de mise à jour peut-être utilisée trois fois sur des espaces réduits - la zone bleue avec les mesures des capteurs 1 et 3, la zone verte avec les mesures du capteur 2, la zone rouge sans mesure - avec à chaque fois un nombre de termes combinatoires moins important ; le coût algorithmique global de la mise à jour est ainsi sensiblement réduit. Grâce à la méthode par partitionnement, des scénarios de surveillance modestes avec un nombre de capteurs et un chevauchement des champs de vue limités peuvent être traités en temps réel avec un filtre PHD multicapteur exact.

## Gestion de capteur

A chaque itération $k$, la gestion de capteurs se déroule en trois phases :

1. prédiction : en fonction du PHD a priori $v_{\Xi_{k+1|k}}$, construire le PHD prédictif $v^u_{\Xi_{k+1|k}}$ pour chaque contrôle possible $u$ ;

2. sélection : en fonction des PHDs prédictifs $v^u_{\Xi_{k+1|k}}$, déterminer le meilleur contrôle selon une *fonction d'objectif* ;

3. contrôle : soumettre les capteurs au contrôle selectionné puis récupérer les mesures courantes $Z_{k+1}$.

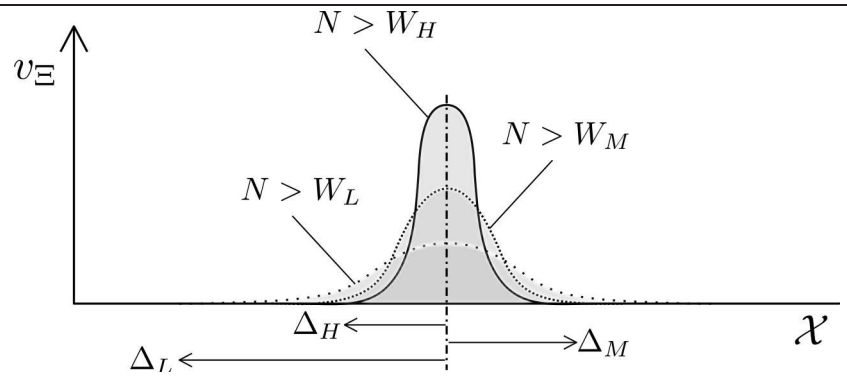
Ce schéma classique de prédiction sur une itération se retrouve dans le contrôleur PENT (Posterior Expected Number of Targets), fruit des premiers travaux sur la gestion de capteur dans le cadre du PHD réalisés par Mahler [Mahl 04].

Une contribution importante de cette thèse est l'extension de la construction du PHD prédictif que l'on retrouve dans le PENT au cas multicapteur. Le principe consiste à extraire du PHD a priori un ensemble de cibles (voir figure plus haut pour une illustration), puis à construire un ensemble de mesures idéales à partir de ces cibles (en omettant les bruits sur l'observation et les fausses alarmes). Enfin, le PHD a priori est mis à jour avec les mesures idéales en entrée *mais pondérées par la probabilité de détection de la cible associée*. En d'autres termes, le PHD prédicif $v^u_{\Xi_{k+1|k}}$ est construit comme la mise à jour du PHD a priori $v_{\Xi_{k+1|k}}$ suivant un processus d'observation simplifié. Dans le cas monocapteur, Mahler obtient une équation de prédiction très proche de l'équation de mise à jour monocapteur, et par conséquent aisément implémentable en temps réel. La construction du cas multicapteur suit la même logique que l'extension de l'équation de mise à jour (6), et peut être simplifiée par une méthode de partitionnement similaire. Pour les mêmes raisons que celles évoquées en section précédente, l'utilisation de l'équation de prédiction exacte n'est envisageable que dans des scénarios modestes.

Conceptuellement, la fonction d'objectif proposée par Mahler pour le contrôleur PENT [Mahl 04] est très simple : le contrôle sélectionné est celui maximisant la prédiction sur le nombre de cibles, c'est à dire encore, par construction du PHD, l'intégrale du PHD prédictif $v^u_{\Xi_{k+1|k}}[1]$. Le principe de ce contrôleur est de favoriser l'observation des zones de l'espace d'état où le poids - c'est à dire l'intégrale du PHD - est elevé, afin de détecter et suivre un maximum de cibles. A travers plusieurs exemples, il est montré dans la thèse que le comportement du PENT n'est pas toujours adapté et qu'il prend parfois de "mauvaises" décisions. Il y a deux raisons principales à cela. Premièrement, la fonction d'objectif ignore par construction l'information contenue dans le PHD concernant la localisation des cibles, puisqu'elle se limite à

déterminer son intégrale sur l'ensemble de l'espace d'état. Elle ne différencie donc pas deux PHDs prédicifs ayant la même intégrale, alors que les cibles potentiellement extractibles de ces deux PHDs peuvent être radicalement différentes. Deuxièmement, le PENT pénalise les contrôles conduisant à une réduction du poids, ce qui n'est pas toujours justifié - par exemple, si le nombre de cible est surestimé dans une zone et qu'une observation est suceptible d'améliorer l'estimation en réduisant le PHD. Par construction, le PHD n'est pas modifié par l'équation de mise à jour dans les zones qui ne sont couvertes par aucun capteur, le PENT a donc tendance à éloigner les capteurs des cibles et ainsi conserver un maximum de poids dans l'espace d'état, plutôt que de "risquer" que le poids ne diminue suite à une observation.

Le contrôleur BET (Balanced Explorer and Tracker) propose une autre approche à la gestion de capteurs, avec une vision plus "opérationnelle" reposant sur la notion de *piste*. Une piste correspond à un certain poids extrait dans une zone limitée de l'espace d'état et peut être vue comme un indice de la présence d'une cible. On distingue trois niveaux de pistes, un niveau plus élevé indiquant une concentration de poids plus importante et donc une plus grande certitude sur la présence d'une cible. Une piste de niveau moyen, par exemple, correspond à un poids supérieur à $W_M$ extrait dans une zone de rayon inférieur à $\Delta_M$ :




Le principe général du BET est de concentrer l'action des capteurs sur les pistes d'intérêt, c'est à dire celles de niveaux faibles, jusqu'à leur disparition - l'origine de la piste est une fausse alarme et aucune nouvelle mesure n'est venue confirmer l'indice de présence d'une cible - ou jusqu'à leur "promotion" vers des niveaux plus élevés - la piste est suffisamment résolue et n'est plus prioriaire. Le seuil délimitant les pistes d'intérêt est fixé différemment pour les zones d'*exploration* et les zones de *poursuite*. Dans les zones d'exploration, où l'objectif est de détecter un maximum de cibles, seules les cibles de niveau bas sont d'intérêt. Dans les zones de poursuite, où la résolution des cibles est favorisée, les cibles de niveau bas et moyen sont d'intérêt. A ce titre, les contrôles sont évalués selon leur capacité à promouvoir des pistes d'intérêt extraites du PHD a priori vers des pistes sans intérêt extraites du PHD

prédictif. Bien entendu, le paramétrage des seuils entre les trois niveaux de pistes est un point critique de la conception du BET.

## Simulation et résultats

Que ce soit dans le cas monocapteur ou multicapteur, les intégrales dans les équations du filtre PHD empêchent leur exploitation directe. Deux techniques d'implémentation semblent être populaires dans le cadre du PHD, les mixtures de Gaussiennes (Gaussian Mixture PHD ou GMPHD) et les méthodes particulaires (Sequential Monte Carlo PHD ou SMCPHD). Une comparaison des deux méthodes [Pace 11] semble montrer que le GMPHD est généralement plus performant, mais son domaine de validité est restreint par des hypothèses plus fortes. Notamment, le GMPHD nécessite une probabilité de détection uniforme sur l'ensemble de l'espace d'état, ce qui est incompatible avec le problème multicapteur. Le filtre PHD multicapteur exact a donc été implémenté par une méthode particulaire en suivant la construction proposée par Vo et al. [Vo 05]. Différents modèles de capteurs et de cibles ont également été conçus et adaptés à l'environnement particulaire afin de pouvoir générer divers scénarios de surveillance.

La première simulation compare le filtre PHD multicapteur exact avec et sans la méthode de partitionnement. Le principe est simple : un même scenario est traité en parallèle avec les filtres résultant des deux méthodes, et une distance de Kullback-Leibler [Aoki 11] évalue la distance entre les deux PHDs obtenus. Les résultats montrent que l'ajout du partitionnement permet de réduire sensiblement le temps d'exécution de la phase de mise à jour, notamment dans les situations critiques - quand les cibles évoluent dans des zones où le recouvrement des champs de vue est fort - où le temps est réduit d'un facteur 100, parfois davantage. Le PHD propagé par les deux méthodes peut être considéré comme identique - l'écart entre les deux restant de l'ordre de $10^{-16}$ tout le long de la simulation - ce qui confirme que le partitionnement simplifie l'exécution de la phase de mise à jour sans pour autant introduire d'erreur.

La seconde simulation compare le filtre PHD multicapteur exact avec le filtre ICA (Iterated-Corrector Approximation) une célèbre approximation multicapteur développée par par Mahler [Mahl 03a, Mahl 10a]. Le principe du ICA est de traiter les différents capteurs *séquentiellement* plutôt que *simultanément* ; c'est à dire, utiliser $S$ fois de suite la mise à jour monocapteur (5) plutôt qu'une fois la mise à jour multicapteur (6). La faiblesse de cette méthode, bien connue, est la dépendance de la solution à l'ordre dans lequel les capteurs sont traités, même si il a été affirmé [Mahl 10a] qu'elle peut être considérée comme négligeable. Un même scénario est testé avec deux configurations de capteurs, une comportant 10 capteurs et l'autre

20. Pour chaque configuration, le meilleur ordre et le pire ordre de traitement par le ICA ont été estimés, le critère étant la distance OSPA moyenne [Vo 08] entre les cibles extraites du PHD et les vraies cibles. On constate d'une part que le filtre exact est de meilleure qualité que le ICA - quel que soit l'ordre - et que la performance du ICA se dégrade sensiblement avec le nombre de capteur. D'autre part, si l'écart entre le meilleur ICA et le pire ICA est relativement faible pour la configuration à 10 capteurs, il augmente fortement avec le nombre de capteur. Ces résultats sont particulièrement intéressants car ils prouvent que, dans certains scénarios du moins, la dégradation de performance du ICA par rapport au PHD de référence est notable et, plus important encore, que l'ordre de traitement des capteurs de l'ICA est un facteur déterminant pour la qualité du filtrage.

La dernière simulation se concentre sur le problème de la gestion de capteurs. Un scénario plus "opérationnel" est généré, dans lequel des routes et des obstacles influencent le déplacement des cibles, et la couverture des capteurs est suffisamment lacunaire pour qu'une gestion des capteurs soit nécessaire. L'estimation de la configuration des cibles est propagée par un filtre PHD exact avec simplification par partitionnement, tandis que la gestion des capteurs est réalisée en parallèle par un contrôleur PENT et un contrôleur BET. La performance des deux contrôleurs est évaluée en comparant la distance OSPA entre les cibles extraites des PHDs propagés et les vraies cibles. Le temps d'exécution de la phase de gestion est également calculé pour les deux contrôleurs à chaque itération. Les résultats montrent clairement que, sur la qualité de l'approximation comme sur le temps d'exécution, le contrôleur BET est nettement supérieur. Comme expliqué dans la section précédente, le PENT a tendance à éloigner les capteurs des zones où le poids est important. Ce phénomène est particulièrement néfaste dans le cadre de l'implémentation particulaire, parce qu'un certain nombre de particules ne sont (presque) jamais observées et se regroupent en nuages se déplaçant dans les zones qui ne peuvent être observées par aucun capteur. En conséquence, le poids augmente considérablement et l'estimation du nombre de cible est largement surévaluée. D'autre part, le BET épargne la phase de prédiction pour certains contrôles potentiels (plus de détails sont donnés dans la thèse), ce qui explique l'amélioration notable du temps d'éxecution. Une autre comparaison avec un contrôle purement aléatoire semble toutefois indiquer que les avantages du BET sont limités. Il est très probable que l'implémentation particulaire proposée soit en partie inadéquate, notamment parce que les particules non couvertes ont tendance à disparaître rapidement à cause du rééchantillonnage. En conséquence, les pistes qui ne sont plus d'intérêt parce que trop résolues sont éliminées dès qu'elles ne sont plus couvertes par les capteurs, alors que leur niveau devraient baisser jusqu'à redevenir éventuellement des pistes d'intérêt et être de nouveau l'objet de nouvelles observations. De façon plus générale, le BET semble incapable d'anticiper la disparition de poids lors du rééchantillonnage, ce phénomène étant propre à l'implémentation

particulaire mais n'ayant pas de mécanisme équivalent dans le cadre théorique du PHD.

## Ouvertures

De nombreuses pistes sont envisageables pour l'approfondissement de cette étude. Sur le plan théorique, le filtre PHD multicapteur exact souffre d'un manque de transparence sur son coût algorithmique. Puisque le temps d'exécution de la phase de mise à jour explose avec le nombre de mesures et/ou de capteurs, un calcul du nombre de termes combinatoires dans la phase de mise à jour (6) permettrait d'évaluer *a priori* le temps d'exécution nécessaire pour une mise à jour exacte. Cela pourrait conduire à un filtre *hybride* qui choisirait, pour chaque élément de partition, si la mise à jour exacte est envisageable compte tenu du nombre de capteurs et de mesures concernés, ou si une approximation de type ICA s'impose.

Sur le plan pratique, une amélioration de l'implémentation particulaire semble nécessaire, notamment pour un bonne exploitation du BET. Une première étape serait d'empêcher la disparition trop rapide des pistes non couvertes, peut être en considérant un rééchantillonnage non systématique [Douc 05]. Indépendamment de cela, le méchanisme de création de particules proposé dans cette thèse n'est pas satisfaisant sur le plan théorique et la recherche d'une fonction d'importance adaptée au problème est une piste à envisager [Rist 10a]. Enfin, une implémentation du PHD multicapteur avec des techniques dérivées du GMPHD relaxant l'hypothèse d'uniformité de la proabilité de détection reste à explorer.

En prenant davantage de recul, d'autres pistes relatives à l'extension du filtre PHD multicapteur apparaissent. En premier lieu, on pourrait envisager l'extension du filtre CPHD (Cardinalized PHD) monocapteur [Mahl 07a] - lui-même une extension du PHD monocapteur propageant la loi de cardinalité du RFS multicible en plus du PHD - au cas multicapteur en s'inspirant des travaux de cette thèse. Un autre sujet d'étude prometteur est la construction d'un filtre propageant le second moment des RFSs multicible, parce que cela permettrait de décrire des systèmes plus complexes comprenant une interaction entre des *paires* de cibles là où les hypothèses du filtre PHD imposent une stricte *indépendance* entre cibles.

L'hypothèse fondamentale sur laquelle repose la validité du PHD comme approximation du RFS, à savoir l'assimilation des RFSs multicible à des Poisson RFSs, est également un domaine d'étude intéressant. Aux yeux de l'auteur, les conséquences de cette hypothèse sur la délimitation de la classe de problèmes pour lesquels le filtrage PHD (ou une méthode dérivée) est valide sont en grande partie inconnues et mériteraient d'être identifiées.

# Contents

# List of Figures

# List of Tables

# List of Algorithms

# Nomenclature

**Acronyms**

BET Balanced Explorer and Tracker

CPHD Cardinalized Probability Hypothesis Density

FISST Finite Set Statistics

FOV Field Of View

GMPHD Gaussian Mixture PHD

HPE Highest Peaks Extractor

ICA Iterated-Corrector Approximation

IS Importance Sampling

JMPD Joint Multitarget Probability Density

JPDA Joint Probabilistic Data Association

KL Kullback-Leibler

MHT Multiple Hypothesis Tracker

NCV Near-Constant Velocity

OSPA Optimal Subpattern Assignment

PA Product Approximation

PENT Posterior Expected Number of Targets

PGF Probability Generating Function

PGFl Probability Generating Functional

PHD Probability Hypothesis Density

PIMS Predicted Ideal Measurement Set

RFS Random Finite Set

SIS Sequential Importance Sampling

SMC Sequential Monte Carlo

SMCPHD Sequential Monte Carlo PHD

SNR Signal-to-Noise Ratio

**Notations**

$\beta[.,\delta_x]$ $\frac{\delta}{\delta x}\beta[.,h]$

$\beta[\delta_Z,.]$ $[\beta[\delta_Z,\bar{g},.]]_{g^{1...S}=0}$

$\beta[\delta_Z,\bar{g},.]$ $\frac{\delta}{\delta Z}\beta[g^1,...,g^S,.]$

$\beta[\delta_\emptyset,.]$ $\beta[0,...,0,.]$

$\beta[\delta_\emptyset,\bar{g},.]$ $\beta[g^1,...,g^S,.]$

$\beta[\delta_{z^j},.]$ $\frac{\delta}{\delta z^i}\beta[g^j,.]$

$\delta_X(.)$ $\sum_{x\in X}\delta_x(.)$

$\frac{\delta}{\delta x}$ $\frac{\partial}{\partial(\delta_x K_{\mathcal{X}})}$

$h^X$ $\prod_{x\in X}h(x)$

$h_X$ $\sum_{x\in X}h(x)$

**Symbols**

$(S_k(p))_{p=1}^{P_k}$ sensor partition (time $k$)

$(S_u(p))_{p=1}^{P_u}$ sensor partition (under control $u$)

$(T_k(p))_{p=0}^{P_k}$ target space partition (time $k$)

$(T_u(p))_{p=0}^{P_u}$ target space partition (under control $u$)

$1_T(.)$ indicator function

$\bigcup_{n=1}^{N} \Xi_n$ union RFS of RFSs $\Xi_1, ..., \Xi_N$

$\bigsqcup_{n=1}^{N} \Xi_n$ joint RFS of RFSs $\Xi_1, ..., \Xi_N$

$\mathcal{C}(Z^1, ..., Z^S)$ set of combinational terms from measurement sets $Z^1, ..., Z^S$

$\chi$ mapping from vectors in $\bigcup_{n=0}^{\infty} \mathcal{X}^n$ to sets in $\mathcal{F}(\mathcal{X})$

$\beta[.]$ cross-term functional

$\beta_p[.]$ reduced cross-term functional (to partition elements $S_k(p)$ and $T_k(p)$)

$\delta_x(.)$ Dirac delta function

$\mathcal{F}(\mathcal{X})$ set of finite subsets of $\mathcal{X}$

$\hat{N}_k$ extracted target number (time $k$)

$\hat{X}_k$ family of extracted targets (time $k$)

$\lambda_{k,k+1}^{b} b_{k,k+1}(.)$ (single-target) birth intensity (time $k+1$)

$\lambda^n$ $n$-th product dimensionless Lebesgue measure

$\lambda_{k,k+1}^{s} s_{k,k+1}(. \mid x)$ (single-target) spawning intensity cond. on state $x$ (time $k+1$)

$\lambda_{k+1}^{c,j} c_{k+1}^{j}(.)$ (single-measurement) false alarm intensity from sensor $j$ (time $k+1$)

$\mathbb{P}$ probability measure (on $\Omega$)

$\mathcal{M}(Z^1, ..., Z^S)$ set of terms from measurement sets $Z^1, ..., Z^S$

$| \, . \, |$ cardinality function

$\mu$ dimensionless reference measure on $\mathcal{X}$

$\Omega$ sample space

$\mathcal{R}$ surveillance region

$\rho_k^j$ (ideal) measurement function of sensor $j$ (time $k$)

$\rho_{\Xi}$ cardinality distribution of RFS $\Xi$

$\Sigma_k(X)$ (multi-measurement) observation RFS cond. on set $X$ (time $k$)

$\varphi_{Z^{j \in J}}(.)$ signature function on measurement sets $Z^1, ..., Z^S$

$\mathcal{X}$ (single-)target state space

$\Xi^T_{k,k+1}(X)$ (multi-target) transition RFS cond. on set $X$ (time $k+1$)

$\Xi^u_{k+1|k+1}$ (multi-target) predictive RFS (under control $u$)

$\Xi_{k+1|k+1}$ (multi-target) posterior RFS (time $k+1$)

$\Xi_{k+1|k}$ (multi-target) predicted RFS (time $k+1$)

$K_{\mathcal{X}}$ unit of the hyper-volume of $\mathcal{X}$

$\mathcal{Z}^j$ observation space of sensor $j$

$K_{\mathcal{Z}^j}$ unit of the hyper-volume of $\mathcal{Z}^j$

$d_{\mathcal{X}}$ Euclidian distance on $\mathcal{X}$

$d_{\mathcal{Z}^j}$ Euclidian distance on $\mathcal{Z}^j$

$F^j_k$ FOV of sensor $j$ (time $k$)

$F^j_u$ FOV of sensor $j$ (under control $u$)

$f^t_{k,k+1}(. \mid x)$ (single-target) transition density cond. on state $x$ (time $k+1$)

$G^{(n)}$ $n$-th order derivative of functional $G$

$G_{\Xi_1,...,\Xi_N}$ joint PGFl of RFSs $\Xi_1, ..., \Xi_N$

$G_{\Xi}$ PGFl of RFS $\Xi$

$J^{(n)}_{\Xi}$ $n$-th order Janossy measure of RFS $\Xi$

$j^{(n)}_{\Xi}$ $n$-th order Janossy density of RFS $\Xi$

$L^{z,j}_k(.)$ (single-target) likelihood from sensor $j$ in measurement $z$ (time $k$)

$m^j_k$ number of current measurements produced by sensor $j$ (time $k$)

$N_k$ estimated target number (time $k$)

$p^s_{k,k+1}(.)$ (single-target) survival probability (time $k+1$)

$P^{(n)}_{\Xi}$ $n$-th order probability distribution of RFS $\Xi$

$p^{d,j}_k(.)$ (single-target) detection probability of sensor $j$ (time $k$)

$p^{d,j}_u(.)$ (single-target) detection probability of sensor $j$ (under control $u$)

$p^{fa,j}_k$ false alarm probability of sensor $j$ (time $k$)

$P_\Xi$ probability distribution of RFS $\Xi$

$p_\Xi$ proability density of RFS $\Xi$

$R_k^u$ reward for control $u$ (time $k$)

$S$ sensor number

$T_u^f$ family of focus tracks covered by control $u$

$T_u^{nf}$ family of non-focus tracks promoted by control $u$

$U_k$ set of available (multi-sensor) controls (time $k$)

$u_k$ selected (multi-sensor) control (time $k$)

$v^u_{\Xi_{k+1|k}}$ (multi-target) predictive PHD for control $u$ (time $k+1$)

$V_\Xi^{(n)}$ $n$-th order factorial moment measure of RFS $\Xi$

$v_\Xi^{(n)}$ $n$-th order product density of RFS $\Xi$

$v_{\Xi_{k+1|k+1}}$ (multi-target) posterior PHD (time $k+1$)

$v_{\Xi_{k+1|k}}$ (multi-target) predicted PHD (time $k+1$)

$V_\Xi$ intensity measure of RFS $\Xi$

$v_\Xi$ intensity (function) of RFS $\Xi$

$X_k$ set of true targets (time $k$)

$x_{i,k}$ $i$-th true target state (time $k$)

$Z_k^j$ set of current measurements produced by sensor $j$ (time $k$)

$z_{i,k}^j$ $i$-th measurement produced by sensor $j$ (time $k$)

$Z_k$ set of current measurements (time $k$)

$Z_{1:k}$ collection of measurements produced up to time $k$

# Introduction

***Curiosity killed the cat...***
***... satisfaction brought it back.***

Proverb

## Motivation

Curiosity may kill the cat indeed, and yet venturing into the unknown - or at least the uncertain - is all the more tempting. Consider Tom, waiting near a road corner for the first cyclists competing in his favourite race, his hands on his brand new binoculars. Here comes the first peloton: Tom focuses his watch on them and counts five cyclists. He is partial to Roy, but sadly he is not in this first group - or so it seems. The frantic cries of other attendants must mean that another peloton is arriving. Tom is eager to switch his watch to this second group, and yet he would like to spend some more time on the five man leading peloton to be sure that they are indeed five - maybe Roy was hidden among them? Tom quickly decides to focus on the second peloton. This one is more loose and is bound to break, a few seconds are long enough to check that Roy is not there. Now Tom would like to look back at the first peloton: where should he focus his binoculars, that is, how much farther have they ridden since he stopped watching them? Switching his focus back and forth between different pelotons seems to be quite challenging, fortunately for Tom the cyclists are bound to ride on the road. What if the cyclists were allowed to wander away from the road? Then Tom could get help from a friend of his with his own binoculars, but how could they coordinate their watch in order to improve their chances to find Roy?

The situation above, albeit simple, arises the main challenges in the *multi-object state estimation problem*. An *observation and identification system* (Tom) is interested in the *states* (position in the race, identity of the cyclist) of some *objects* (the cyclists) evolving in a bounded *region* (the corner of the road next to Tom). The

state of each object is uncertain - otherwise the race would lack any challenge! - but the system can rely on *measurements* (a glance at the road through the binoculars) produced by *sensors* (the binoculars). These measurements are usually *noisy* (Tom cannot see very well because of the sun reflection), yet *sequential* observations allows the system to build an *estimation* of the *true situation* (the real position of the cyclists on the corner of the road), modified *dynamically* following each new observation. The system can eventually make decisions to *control* the sensor's action (Tom may choose to focus his googles on another point of the road) according to an *objective* (spotting Roy among all the cyclists). The system may rely on measurements from several sensors (Tom's friend with his own binoculars), in this case the measurements must be *shared* (his friend thoroughly describes what he sees to Tom) so that the system can produce a single estimation based on sequential *multi-sensor observations*. Finally, the system must be able to produce *multi-sensor controls* (Tom should provide instructions to his friend in order to coordinate their watch).

Many concrete problems fit through the state estimation framework given above, or at least share some of its salient features, in various fields such as econometric [Yell 10], biomedical engineering [Juan 09], meteorology [Solt 11] and of course tracking [Gust 02]. The best known is perhaps the target detection and/or tracking problem in surveillance activities, whether civilian or military, because it shares all of the features above: the *targets* (i.e. the objects) are usually moving in a region of unknown topography and their number is time-varying, the sensor coverage is lacking (i.e. the sensors cannot cover all the surveillance region simultaneously) hence decisions must be made by the sensor manager, the sensors can miss a target or produce a false alarm, etc. In a world where both the targets and the sensors are of increasing complexity, the improvement of the surveillance activities is a growing concern and a challenging problem.

## The multi-sensor/multi-target filtering problem

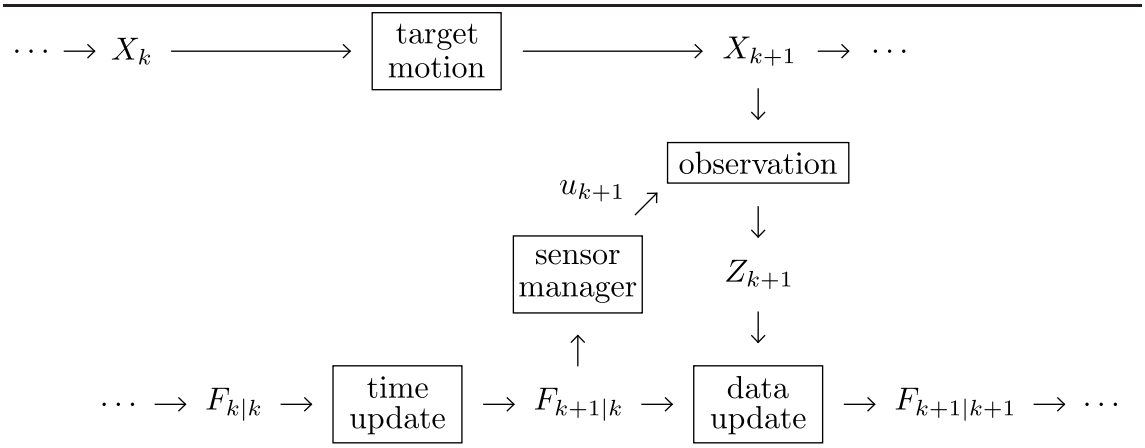
The scope of this thesis fits in the general *multi-sensor/multi-target filtering problem* whose aim is to estimate the states of a varying number of targets through sequential observations by several sensors. The surveillance region $\mathcal{R} \subset \mathbb{R}^2$ is a bounded ground region - e.g. the whereabouts of a military base. The targets are ground objects evolving in the surveillance region, their state $x$ is a four-variable vector, two for position and two for velocity coordinates, belonging to the *target state space* $\mathcal{X} \subset \mathbb{R}^4$ equipped with the Euclidian distance $d_{\mathcal{X}}$. For description purposes, targets may be labeled in an arbitrary order. Likewise, their state may be labeled on a time scale with an arbitrary origin. In the general case, $x_{i,t}$ will denote the state of the $i$-th target at time $t$, but the target label and/or time subscript may be omitted

when irrelevant.

The surveillance region is observed by $S$ sensors. Each sensor $j$ has its own characteristics (probability of detection, false alarm rate, measuring accuracy, etc.) and produces measurements $z^j$ belonging to its own observation space $\mathcal{Z}^j \subset \mathbb{R}^{d^j}$ equipped with the euclidian distance $d_{\mathcal{Z}^j}$. As for the targets, $z^j_{i,t}$ will denote the $i$-th measurement produced by the sensor $j$ at time $t$, but the origin sensor superscript, the measurement label and/or the time subscripts may be omitted when irrelevant. Each sensor $j$ has limited coverage in the sense that it may not focus on the whole state space simultaneously. Its *field of view* (FOV) $F^j_t \subset \mathcal{X}$ is such that targets cannot be detected by sensor $j$ at time $t$ unless they belong to its FOV $F^j_t$. The FOV of each sensor can be modified to some extent by *dynamically managing* the sensor, i.e. providing instructions to the sensor on the region of the state space it should focus on based on the current estimation of the target number and their localization. One of the trickiest part of the multi-sensor/multi-target filtering problem is to design a proper *objective* for the surveillance activity and be able to manage the sensors accordingly.

Because the multi-target/multi-sensor filtering problem is inherently dynamic, a common time scale is required for the targets' model, the observation process and the control process. For simplicity's sake, the time is discretized in (time) steps. The data flow over a time step can be depicted as follows:

**Figure 1** Data flow of the filtering process (time $k$)




where:

- $X_k$ is the collection of all target states at time (or step) $k$ (the 'true situation');
- $Z_k$ is the collection of all measurements produced at time $k$, regardless of their origin sensor;

- $F_{k|k}$ (resp. $F_{k+1|k}$) is the filtered state at time $k$ (resp. $k+1$) based on the measurements produced up to time $k$;

- $u_{k+1}$ is the multi-sensor control produced at time $k+1$.

The data flow (figure 1) implies that several important assumptions about the system are made:

- one can find a time step adapted to both target and sensor dynamics;

- the sensor system is *centralized* and *without delay*;

- the sensors are *synchronized*.

Arguably, the first assumptions depends chiefly on the sensor characteristics. Since the scope of this thesis is limited to ground targets whose typical time step may last a few seconds, the assumption that the ground surveillance sensors (typically, radars or cameras) are able to produce measurements every time step seems reasonable enough.

The sensor system being centralized is a valuable convenience for the design of the filter, because it implies that all the measurements produced in a given time step (say, $k+1$) are immediately available for the data update (see figure 1) occuring during the same step ($F_{k+1|k} \rightarrow F_{k+1|k+1}$). Notably, this implies that there is delay in neither the observation nor the data transmission processes. Likewise, as soon as the sensor manager produces the selected multi-sensor control $u_{k+1}$, the sensors are controlled accordingly and the collection of current measurement $Z_{k+1}$ is available instantaneously.

The assumption on the synchronization of the sensors is equally important, because it has two major consequences. The desirable effect is that it significantly simplifies the design of the sensor manager, which can be synchronized on the same time step as the targets (see figure 1): the sensor manager selects a *single* multi-sensor control $u_{k+1}$ every time step, providing instructions to *all the sensors simultaneously*. The other consequence, though not visible on the data flow, is the absence of temporality between the elements in the current collection of measurements $Z_{k+1}$ - that is, the element order in $Z_{k+1}$ is arbitrary and cannot be used to set an order of precedence among the measurements produced by a given sensor *or* among the sensors. This consequence is critical to the design of the data update step since it implies that one cannot hope to update the filtered state by proceeding *sequentially* with the measurements from the different sensors, rather, one must deal with the sensors *all at once*.

## Organization of the thesis

The thesis is organized as follows:

**Chapter 1** provides the main theoretical tools that are required for the design of the *probability hypothesis density* (PHD) filter. The chapter focuses mainly on the *random finite set* (RFS) theory, but also introduces the single-sensor PHD filter and some metrics adapted to the RFS framework.

**Chapter 2** describes thoroughly the construction of the multi-sensor PHD filter. The first part provides an adaptation of Mahler's construction of the single-sensor case [Mahl 03a] which is important to grasp the rigorous extension to the multi-sensor case that follows. A joint partitioning of the state space and the sensors is then proposed in order to simplify the construction of the multi-sensor PHD filter without approximation. Based on this new reference, common multi-sensor approximations are then discussed. The extension to the multi-sensor case as well as the partition method are key contributions of the thesis.

**Chapter 3** deals with the sensor management problem. The first part focuses on the target extraction process, then on the rigorous extension of Mahler's *predicted ideal measurement set* (PIMS) [Mahl 04] to the multi-sensor case and its simplification by a similar partitioning as in chapter 2. It then focuses on the design of a sensor manager. Mahler's *posterior expected number of targets* (PENT) manager [Mahl 04] is introduced and analyzed through simple situations where it seems inadequate. A new solution based on 'operational' objective is then proposed, the *balanced explorer and tracker* (BET) manager. The extension to the multi-sensor case, the partitioning and the BET manager are other important contributions of this thesis.

**Chapter 4** first describes the modelization of surveillance scenarii, then the implementation of the multi-sensor PHD filter and sensor manager through *sequential Monte Carlo* (SMC) techniques, and finally the main results obtained on simulated data.

All the mathematical proofs pertaining to chapters 1, 2, and 3 may be found in **appendix A**, and a brief description of the principles of *importance sampling* (IS) and *sequential importance sampling* (SIS) is provided in **appendix B**.

CHAPTER 1

# Background

MAhler should take the credit for much of the original work on the RFS theory [Mahl 02], since he established grounds for the first rigorous Bayesian filter in a multi-object context with the *finite set statistics* (FISST) approach. In later works, Mahler also proposed the construction of the PHD filter [Mahl 03a], an approximation of the more general multi-objet Bayesian filter leading to a tractable single-sensor/multi-target filter. Not surprinsingly, the new set-based approach to multi-target tracking arouse some interest in the tracking community and many implementations of PHD-based multi-target tracking filters have been proposed [Maeh 06, Pham 07, Juan 09]. But, more generally, the random finite sets and their derivatives proved to be an exciting field of study and have been the topic of several recent theses [Viho 04, Clar 06, Tobi 06, Pant 07, Vo 08]. This chapter describes the basic notions about random set theory that were needed for the design of the PHD filter in the single-sensor case, and for its rigorous extension to the multi-sensor case as well. The PHD filter is also briefly described, its construction being fully detailed in the next chapter.

## 1.1 Random finite sets

Even though the early work regarding the random finite set theory can be found in Mahler's work [Mahl 02, Mahl 03a], this section is mainly an adaptation of Vo's thesis [Vo 08] which provides a well-written summary of the essential definitions and properties pertaining to the RFS.

### 1.1.1 Definition

Essentially, as it will be formally established later in section 1.2, the purpose of the set-based filtering approach is to consider the collection of true targets $X_k = \{x_{1,k}, x_{2,k}, \ldots, x_{N_k,k})\}$) as the realization of a random variable and build a filtered state - that is, an estimation of this 'true situation' - upon the knowledge of this random variable acquired through successive observations rather than upon several maintained tracks. Because the realization of this random variable must cover every possible number of targets and, for any target number, every possible combination of target states, each realization belongs to $\mathcal{F}(\mathcal{X})$, the set of all the finite subsets of the state space $\mathcal{X}$.

**Definition 1.1.** *A RFS $\Xi$ defined on $\mathcal{X}$ is a measurable mapping [Vo 08]:*

$$\begin{aligned} \Xi \ : \ & \Omega \to \mathcal{F}(\mathcal{X}) \\ & \omega \mapsto X = \Xi(\omega) \end{aligned} \tag{1.1}$$

*where $(\Omega, \sigma(\Omega), \mathbb{P})$ is a probability space equipped with the Matheron topology.*

As usual in the study of random variables, one's focus shifts easily from the probability measure $\mathbb{P}$ - defined on the sample space $\Omega$ - to the more practical probability distribution of the RFS, i.e. the probability measure $P_\Xi$ defined on $\mathcal{F}(\mathcal{X})$ by [Vo 08]:

$$P_\Xi(\mathcal{T}) \stackrel{def}{=} \mathbb{P}(\{\omega \in \Omega : \ \Xi(\omega) \in \mathcal{T}\}) = \mathbb{P}(\{X \in \mathcal{T}\}) \tag{1.2}$$

for any Borel subset $\mathcal{T}$ of $\mathcal{F}(\mathcal{X})$. Like any random variable, $\Xi$ is completely described by its probability distribution $P_\Xi$. From now on, functions, subsets and events are assumed to be measurable or Borel whenever appropriate.

Intuitively, a RFS is well adapted to the description of a process producing different point patterns with associated probabilities. Actually, RFSs can be seen as particular cases of more general objects called *point processes* and many results presented in this section can be found in their point process equivalent in [Sing 09]. Although point processes will not be referred to anymore later in this thesis, a brief description of the similarities between the two notions provides another approach to grasp the concept of RFS. The following definitions concerning the point process are given in [Vo 08]:

1. a *counting measure* $n$ on $\mathcal{X}$ is a measure taking values in $\mathbb{N} \cup \{\infty\}$ such that $n(\mathcal{T})$ is finite for any bounded subset $\mathcal{T}$ of $\mathcal{X}$;

2. a counting measure $n$ is *simple* if $n(\{x\}) \in \{0, 1\}, \forall x \in \mathcal{X}$;

3. a counting measure $n$ is *finite* if $n(\mathcal{X}) < \infty$;

4. a *point process* $N$ on $\mathcal{X}$ is a measureable mapping from a sample space $\Omega$, where $(\Omega, \sigma(\Omega), \mathbb{P})$ is a probability space, to the space of counting measures on $\mathcal{X}$;

5. a point process $N$ is *simple* if $N(\omega)$ is simple almost surely;

6. a point process $N$ is *finite* if $N(\omega)$ is finite almost surely;

7. a point process $N$ is *simple-finite* if it is simple and finite.

Any simple-finite point process $N$ on $\mathcal{X}$ may be associated with an equivalent RFS $\Xi$ on $\mathcal{X}$ bound by the following relation:

$$\forall \mathcal{T} \subseteq \mathcal{X},\ N(\omega)(\mathcal{T}) = |\Xi(\omega) \cap \mathcal{T}| \tag{1.3}$$

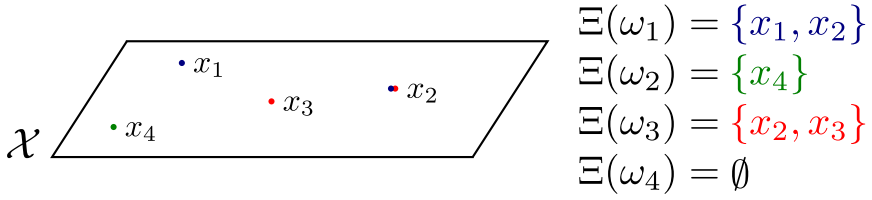
where $|.|$ is the cardinality function. This important relation shows that a RFS on $\mathcal{X}$ can be completely described by counting the occurences of points falling into subspaces of $\mathcal{X}$. A recent book of Streit [Stre 10] thoroughly describes the detection and tracking problem from the point processes' point of view; not surprisingly, many notions are quite similar to those developed in the RFS framework.

**Example 1.1.** *One can define a RFS $\Xi$, with the sample space $\Omega = [0, 1]$ equipped with the uniform probability, whose distribution probability is described by:*

$$\Xi(\omega) = \begin{cases} \{x_1, x_2\} & if \quad 0 \leqslant \omega < \dfrac{1}{2} \\ \{x_2, x_3\} & if \quad \dfrac{1}{2} \leqslant \omega < \dfrac{2}{3} \\ \{x_4\} & if \quad \dfrac{2}{3} \leqslant \omega < \dfrac{3}{4} \\ \{\emptyset\} & if \quad \dfrac{3}{4} \leqslant \omega \leqslant 1 \end{cases} \tag{1.4}$$

*This RFS may be depicted as follows:*

**Figure 1.1** Illustration of a RFS



*Equivalently, one can define a simple-finite point process $N$, with the same probability space, whose distribution probability is built as follows:*

$$N(\omega) = \begin{cases} n: & n(\{x_1\}) = 1, n(\{x_2\}) = 1, n(\mathcal{X}) = 2 & if \quad 0 \leqslant \omega < \frac{1}{2} \\ n: & n(\{x_2\}) = 1, n(\{x_3\}) = 1, n(\mathcal{X}) = 2 & if \quad \frac{1}{2} \leqslant \omega < \frac{2}{3} \\ n: & n(\{x_4\}) = 1, n(\mathcal{X}) = 1 & if \quad \frac{2}{3} \leqslant \omega < \frac{3}{4} \\ n: & n(\mathcal{X}) = 0 & if \quad \frac{3}{4} \leqslant \omega \leqslant 1 \end{cases} \tag{1.5}$$

The point process formulation (1.5) also provides some insight on the construction of the PHD. Since there is no addition operator defined on $\mathcal{F}(\mathcal{X})$ - $\{x_1, x_2\} + \{x_2, x_3\}$ has no mathematical sense, the "classical" expectation operator cannot be applied to the RFS - $\mathbb{E}[\Xi]$ has no sense either. However, additioning counting measures has a mathematical sense, and so does the expectation $\mathbb{E}[N]$. Applying this expectation on the point process described in equation (1.5) provides the following description of the RFS $\Xi$: "on average, there are 0.5 targets located in $x_1$, 0.66 targets in $x_2$, 0.17 targets in $x_3$, 0.08 targets in $x_4$ and none elsewhere". Propagating this information through time is, in a nutshell, the principle of the PHD filter.

Note that a RFS is equivalent to a *simple*-finite point process; the "simple" qualification hence precludes the possibility of repeated elements in any realization of a RFS. In the multi-target tracking framework, this implies that a RFS-based multi-target representation does not cover the possibility that several targets share the *exact same* state. Given the usual nature of target states (position and/or velocity coordinates in a "large" surveillance region), this restriction may seem to be of little importance, even though this could have some undesirable effects in the tracking of well organized targets moving together (in a convoy for example). Note also that, since the realization of a RFS is a set of points, these points are *unordered* and any labelization is necessarily arbitrary. As it will be seen later in section 1.2, this has direct consequences on the design of multi-target tracking filters.

Yet another equivalent description of a RFS $\Xi$ is given by the following distributions [Vo 08]:

- a cardinality distribution $\rho_\Xi$;
- a family of symmetric probability distributions $\{P_\Xi^{(n)}\}_{n \in \mathbb{N}}$.

where $\rho_\Xi$ describes the distribution of the number of elements in any realization $X$ of the RFS and $P_\Xi^{(n)}$ describes the joint spatial distribution of these elements in the state space $\mathcal{X}$. Note that the probability distributions $P_\Xi^{(n)}$ are symmetric since the

element order in a set is arbitrary.

The *union RFS* of a finite number of RFSs can be defined as follows (adapted from [Vo 08]):

**Definition 1.2.** *For any* $n \in [1\ N]$, *let* $\Xi_n : \Omega \to \mathcal{F}(\mathcal{X})$ *be a RFS. Then, the union RFS* $\bigcup_{n=1}^{N} \Xi_n$ *is the RFS* $\Xi : \Omega^N \to \mathcal{F}(\mathcal{X})$ *given by:*

$$\forall \omega = (\omega_1, ..., \omega_N) \in \Omega^N,\ \Xi(\omega) \stackrel{def}{=} \bigcup_{n=1}^{N} \Xi_n(\omega_n) \tag{1.6}$$

Finally, an extension to the union RFS can be defined as follows:

**Definition 1.3.** *For any* $n \in [1\ N]$, *let* $\Xi_n : \Omega \to \mathcal{F}(\mathcal{X}_n)$ *be a RFS. Then, the joint RFS* $\bigsqcup_{n=1}^{N} \Xi_n$ *is the RFS* $\Xi : \Omega^N \to \bigsqcup_{n=1}^{N} \mathcal{F}(\mathcal{X}_n)$ *given by:*

$$\forall \omega = (\omega_1, ..., \omega_N) \in \Omega^N,\ \Xi(\omega) \stackrel{def}{=} \bigsqcup_{n=1}^{N} \Xi_n(\omega_n) \tag{1.7}$$

The joint RFS can be seen as a "disjoint union" of a finite number of RFSs defined on *different* spaces. This definition was added by the author in order to describe the multi-sensor observation process as the "union" of single-sensor processes (see chapter 2). In particular, the following property will be useful in the construction of the multi-sensor PHD equations:

**Property 1.1.** *For any* $n \in [1\ N]$, *let* $\Xi_n : \Omega \to \mathcal{F}(\mathcal{X}_n)$ *be a RFS on* $\mathcal{X}_n$. *If the RFSs* $\Xi_i$ *are statistically independent, then, for any family of subsets* $\{\mathcal{T}_n\}_{n=1}^{N}, \mathcal{T}_n \subseteq \mathcal{F}(\mathcal{X}_n)$*:*

$$P_{\bigsqcup_{n=1}^{N} \Xi_n}\left(\bigsqcup_{n=1}^{N} \mathcal{T}_n\right) = \prod_{n=1}^{N} P_{\Xi_n}(\mathcal{T}_n) \tag{1.8}$$

The proof is given in appendix A.

### 1.1.2 Probability density

The notions of measure, integration and probability density are equally important for the construction of the RFS-based filter. In RFS theory, the usual reference measure on $\mathcal{F}(\mathcal{X})$ is the dimensionless measure $\mu$ given by [Vo 05, Vo 08]:

$$\mu(\mathcal{T}) \overset{def}{=} \sum_{n=0}^{\infty} \frac{\lambda^n(\chi^{-1}(\mathcal{T}) \cap \mathcal{X}^n)}{n!} \tag{1.9}$$

for any subset $\mathcal{T}$ of $\mathcal{F}(\mathcal{X})$, where $\lambda^n$ is the $n$-th product dimensionless Lebesgue measure on $\mathcal{X}^n$, and $\chi$ is a mapping on vectors defined on Cartesian product spaces $\mathcal{X}^n$ and $n$-element set spaces defined by:

$$\begin{aligned} \chi \; : \; & \bigcup_{n=0}^{\infty} \mathcal{X}^n \to \mathcal{F}(\mathcal{X}) \\ & x_1, ..., x_n \mapsto \{x_1, ..., x_n\} = \chi(x_1, ..., x_n) \end{aligned} \tag{1.10}$$

Intuitively, in order to measure a subset $\mathcal{T}$, one must stack its elements by size, measure each element with the proper dimensionless measure according to its size, then sum them to obtain the size of $\mathcal{T}$. Note that for each $n$-element set belonging to $\mathcal{T}$, there is $n!$ corresponding elements belonging to the Cartesian product $\mathcal{X}^n$ with the same measure, hence the factorial on the denominator. Because the hyper-volume in the state space $\mathcal{X}$ may have unit (e.g. $m^4.s^{-2}$ for a 2D surveillance region with position and velocity coordinates), a dimensionless measure on Cartesian products is required to keep the homogeneity between the measures of vectors belonging to different Cartesian product spaces.

**Example 1.2.** *Consider the one-variable state space $\mathcal{X} = [0\ 1]$ - the state of point being, for example, its position on the real unit segment. Let us consider the following subset $\mathcal{T} \subset \mathcal{F}(\mathcal{X})$:*

$$\mathcal{T} \overset{def}{=} \begin{cases} \{\{x\} \mid x \in [0\ 1/2]\} \\ \{\{x, y\} \mid x \in [0\ 1/3], y \in [0\ 1]\} \end{cases} \tag{1.11}$$

*then $\chi^{-1}(\mathcal{T})$ can be decomposed as follows:*

$$\chi^{-1}(\mathcal{T}) : \begin{cases} \chi^{-1}(\mathcal{T}) \cap \mathcal{X} = [0\ 1/2] \\ \chi^{-1}(\mathcal{T}) \cap \mathcal{X}^2 = [0\ 1/3] \times [0\ 1] \cup [0\ 1] \times [0\ 1/3] \\ \chi^{-1}(\mathcal{T}) \cap \mathcal{X}^n = \emptyset, n \notin \{1, 2\} \end{cases} \tag{1.12}$$

*thus we have:*

$$\begin{cases} \lambda(\chi^{-1}(\mathcal{T}) \cap \mathcal{X}) = 1/2 \\ \dfrac{\lambda^2(\chi^{-1}(\mathcal{T}) \cap \mathcal{X}^2)}{2} = 1/3 \times 1 \\ \dfrac{\lambda^n(\chi^{-1}(\mathcal{T}) \cap \mathcal{X}^n)}{n!} = 0, n \notin \{1, 2\} \end{cases} \tag{1.13}$$

*which leads to:*

$$\mu(\mathcal{T}) = \lambda(\chi^{-1}(\mathcal{T}) \cap \mathcal{X}) + \frac{\lambda^2(\chi^{-1}(\mathcal{T}) \cap \mathcal{X}^2)}{2} = \frac{5}{6} \tag{1.14}$$

The construction of the measure on $\mathcal{F}(\mathcal{X})$ is naturally followed by the definition of the integral of a function $f : \mathcal{F}(\mathcal{X}) \to \mathbb{R}$ over a subset $\mathcal{T}$ [Vo 05, Vo 08]:

$$\begin{aligned}\int_{\mathcal{T}} f(X)\mu(dX) &\overset{def}{=} \sum_{n=0}^{\infty} \frac{1}{n!} \int_{\mathcal{X}^n} 1_{\mathcal{T}}(\chi(x_1, ..., x_n)) f(\chi(x_1, ..., x_n)) \lambda^n(dx_1...dx_n) \\ &= \sum_{n=0}^{\infty} \frac{1}{n!} \int_{\chi^{-1}(\mathcal{T}) \cap \mathcal{X}^n} f(\{x_1, ..., x_n\}) \lambda^n(dx_1...dx_n)\end{aligned} \tag{1.15}$$

Note that the notation $dx_1$ is used in standard integrals, hence $dx_1$ may have unit depending on the state space $\mathcal{X}$. However, recall that $\lambda^n(dx_1...dx_n)$ is dimensionless. If $K_{\mathcal{X}}$ is the unit of the hyper-volume of $\mathcal{X}$, then:

$$\forall n \in \mathbb{N},\ dx_1...dx_n = \lambda^n(dx_1...dx_n) K_{\mathcal{X}}^n \tag{1.16}$$

Then, the probability density $p_{\Xi}$ of a RFS $\Xi$, if it exists, is given by the Radon-Nikodým derivative of the probability distribution $P_{\Xi}$ with respect to the measure $\mu$ [Nguy 06, Vo 08]:

$$\forall \mathcal{T} \subseteq \mathcal{F}(\mathcal{X}),\ P_{\Xi}(\mathcal{T}) = \int_{\mathcal{T}} p_{\Xi}(X)\mu(dX) \tag{1.17}$$

### 1.1.3 Janossy measures and Janossy densities

The measuring process on $\mathcal{F}(\mathcal{X})$ is by no means trivial. Because the different elements in $\mathcal{F}(\mathcal{X})$ are sets of various size, the "trick" is to stack these elements by size and then, for each size $n$, "go back" in the well known associated product space $\mathcal{X}^n$ - using the $\chi$ function given by equation (1.10) - where standard Lebesgue measures are available. Thus, its seems fairly natural to "split" the measure on $\mathcal{F}(\mathcal{X})$ by a family of measures on product spaces $\mathcal{X}^n, n \in \mathbb{N}$, encapsulating the more intuitive notions of cardinality distribution $\rho_{\Xi}$ and spatial distribution $P_{\Xi}^{(n)}$ defined in section 1.1.1. Indeed, one can defined the family of *Janossy measures* $\{J_{\Xi}^{(n)}\}_{n \in \mathbb{N}}$ [Vo 08]:

**Definition 1.4.** *The n-th order Janossy measure* $J_\Xi^{(n)}$ *of a RFS* $\Xi$ *is the measure on* $\mathcal{X}^n$ *given by:*

$$J_\Xi^{(n)}(.) \overset{def}{=} n!\rho_\Xi(n)P_\Xi^{(n)}(.) \tag{1.18}$$

*where* $\rho_\Xi$ *is the cardinality distribution,* $P_\Xi^{(n)}$ *is the n-th order probability measure of RFS* $\Xi$*.*

*If* $J_\Xi^{(n)}$ *admits a density (with respect to the standard Lesbegue measure on* $\mathcal{X}^n$*), then it is called the n-th order Janossy density* $j_\Xi^{(n)}$ *(*$j_\Xi^{(0)}K_\mathcal{X} = J_\Xi^{(0)}$ *by convention).*

Note that the Janossy densities vanish when evaluated on identical points (e.g. $j_\Xi^{(n)}(x_1, x_1, x_2, ...) = 0$), since there is no repeated points in any realization of a RFS. One must be careful not to confuse the probability density $p_\Xi$ with the Janossy densities $\{j_\Xi^{(n)}\}_{n\in\mathbb{N}}$. While the probability density $p_\Xi$ is defined on $\mathcal{F}(\mathcal{X})$ and covers all the possible realizations of the RFS $\Xi$, the Janossy density $j_\Xi^{(n)}$ covers the possible realizations of the RFS $\Xi$ *among those with n elements only* and is *not* a probability density since $\int_{\mathcal{X}^n} j_\Xi^{(n)}(X)dX \neq 1$ in the general case. Furthermore, $p_\Xi$ is *dimensionless* since it is defined with respect to the dimensionless measure $\mu$ on $\mathcal{F}(\mathcal{X})$, while $j_\Xi^{(n)}$ has unit $K_\mathcal{X}^{-n}$ since it is defined with respect to the standard measure on $\mathcal{X}^n$. However, these densities are related through the following equality [Vo 08]:

$$p_\Xi(\{x_1, ..., x_n\}) = j_\Xi^{(n)}(x_1, ..., x_n)K_\mathcal{X}^n \tag{1.19}$$

The proof is given in appendix A. Furthermore, since the spatial distribution $P_\Xi^{(n)}$ can be recovered with the relation $P_\Xi^{(n)}(.) = J_\Xi^{(n)}(.)/J_\Xi^{(n)}(\mathcal{X}^n)$ and the cardinal distribution $\rho_\Xi$ with the relation $\rho_\Xi(n) = J_\Xi^{(n)}(\mathcal{X}^n)/n!$, the family of Janossy measures $\{J_\Xi^{(n)}\}_{n\in\mathbb{N}}$ describes completely the RFS $\Xi$. Clearly the cardinal and the spatial distribution are more intuitive because they embody the algorithmic process for the sampling of a RFS - first draw a number of element according to $\rho_\Xi$, then distribute the elements in state space by sampling from the appropriate $P_\Xi^{(n)}$. However, the Janossy notations are better adapted to set-based calculus.

### 1.1.4 Factorial moments

Like usual random variables, RFS can be characterized through their moments. The *n-th order factorial moment measure* $V_\Xi^{(n)}$ of a RFS $\Xi$ can be defined as follows [Vo 08]:

**Definition 1.5.** *Let* $\Xi$ *be a RFS,* $\{T_n\}_{n\in\mathbb{N}}$ *a family of subsets of* $X$*. Then, the n-th order factorial moment measure* $V_\Xi^{(n)}$ *is defined by:*

$$V_\Xi^{(n)}(T_1 \times ... \times T_n) \overset{def}{=} \mathbb{E}\left[\sum_{x_1\neq...\neq x_n\in\Xi(\omega)} 1_{T_1\times...\times T_n}((x_1, ..., x_n))\right] \tag{1.20}$$

*The first order factorial moment measure $V_\Xi$ ($V_\Xi^0$ by convention) is commonly known as the intensity measure:*

$$V_\Xi(T_1) \stackrel{def}{=} \mathbb{E}\left[\Xi(\omega) \cap T_1\right] \tag{1.21}$$

*If it exists, the n-th order product density is the function $v_\Xi^{(n)}$ defined on $\mathcal{X}^n$ such that:*

$$V_\Xi^{(n)}(T_1 \times ... \times T_n) = \int_{T_1} ... \int_{T_n} v_\Xi^{(n)}(x_1, , ..., x_n) dx_1 ... dx_n \tag{1.22}$$

*The first order factorial product density $v_\Xi$ ($v_\Xi^0$ by convention) is commonly known as the intensity (function):*

$$V_\Xi(T_1) = \int_{T_1} v_\Xi(x) dx \tag{1.23}$$

The notion of moment of a RFS is perhaps more difficult to grasp as the notion of probability, especially when the subsets $T_i$ are not disjoints. Intuitively, the quantity $p_\Xi(\{x_1, ..., x_n\}) K_\mathcal{X}^n dx_1 ... dx_n$ can be seen as the probability that the realization $X$ of the RFS has *exactly* $n$ points, each point in a different neighborhood $dx_i$. On the other hand, the quantity $v_\Xi^{(n)}(x_1, ..., x_n) dx_1 ... dx_n$ can be seen as the probability that $X$ of the RFS has *at least* $n$ points in the different neighborhoods $dx_i$ [Vo 08]. For more details, a comparison between Janossy and factorial moment measures can be found in [Vere 88] (pp.133 - 134). Easily enough, the design of the PHD filter focuses on the first moment only, whose formulation given by (1.21) is rather suggestive: the first moment $V_\Xi(T)$ counts the average number of points falling in the subset $T \subseteq \mathcal{X}$.

Note that if $dx_1, ..., dx_N$ are infinitesimal disjoint neighborhoods, then using definitions 1.4 and 1.5 yields:

$$V_\Xi^{(N)}(dx_1 \times ... \times dx_N) = \sum_{n=0}^{\infty} \frac{1}{n!} \underbrace{\int \cdots \int}_{\mathcal{X}^n} J_\Xi^{(n+N)}(dx_1, ..dx_N, dx_{N+1}, ..., dx_{N+n}) \tag{1.24}$$

Note also that the $n$-th order product density $v_\Xi^{(n)}$, similarly to the $n$-th order Janossy density $j_\Xi^{(n)}$, is defined on product space $\mathcal{X}^n$ and has unit $K_\mathcal{X}^{-n}$. In fact, product densities $v_\Xi^{(n)}$ are "closer" to the Janossy densities $j_\Xi^{(n)}$ than to the probability density $p_\Xi$ in the sense that their family comes as a "toolbox", each tool being adapted on a standard product space $\mathcal{X}^n$, rather than as a single yet more complicated tool adapted for the whole space $\mathcal{F}(\mathcal{X})$.

Interestingly, Mahler proposed in [Mahl 03a] a dimensionless "multitarget moment density" $D_\Xi$, defined on $\mathcal{F}(\mathcal{X})$ by:

$$\forall X \in \mathcal{F}(\mathcal{X}),\ D_\Xi(X) \stackrel{def}{=} \int_{\mathcal{F}(\mathcal{X})} p_\Xi(X \cup W) \mu(dW) \tag{1.25}$$

Table 1.1: Relations between different densities

| Space | $\mathcal{X}^n$ | $\mathcal{F}(\mathcal{X})$ |
|---|---|---|
| Probabilities (unit) | $j_\Xi^{(n)}(x_1,...,x_n)$ $(K_\mathcal{X}^{-n})$ | $p_\Xi(\{x_1,...,x_n\})$ (none) |
| Moments (unit) | $v_\Xi^{(n)}(x_1,...,x_n)$ $(K_\mathcal{X}^{-n})$ | $D_\Xi(\{x_1,...,x_n\})$ (none) |

Equation (1.25) is perhaps more suggestive than those of the factorial moments: the moment $D_\Xi(\{x_1,...,x_n\})$ gives the probability that the realization of the RFS contains *at least* these points. In parallel to equation (1.19) we can write:

$$D_\Xi(\{x_1,...,x_n\}) = v_\Xi^{(n)}(x_1,...,x_n)K_\mathcal{X}^n \tag{1.26}$$

However, because Mahler defines the PHD $D_\Xi$ as the intensity (i.e. the first order product density) rather than the multitarget moment density (see definition 1 in [Mahl 03a]), the equivalence between the two notions - $\forall x \in \mathcal{X},\ D_\Xi(\{x\}) = D_\Xi(x)$ - must be explicitely proven (see theorem 2 in [Mahl 03a]).

### 1.1.5 Probability generating functionals

#### Definition and fundamental properties

The notion of *probability generating functional* (PGFl) is central to the construction of the PHD [Mahl 03a]. Conceptually, the PGFl may be seen as a generalization of the *belief-mass functional* [Vo 08]:

**Definition 1.6.** *The belief-mass functional $\beta_\Xi$ of RFS $\Xi$ is the function given by:*

$$\beta_\Xi(T) \overset{def}{=} \mathbb{P}(\{\Xi(\omega) \subseteq T\}) = \mathbb{E}[1_T^{\Xi(\omega)}] \tag{1.27}$$

*where $T$ is any subset of $\mathcal{X}$, $1_T(.)$ is the indicator function defined on $\mathcal{X}$ and, for any realization $X = \Xi(\omega)$, $1_T^X = \prod_{x\in X} 1_T(x)$. Besides, if $\Xi$ admits a probability density:*

$$\beta_\Xi(T) = \int_{\mathcal{F}(\mathcal{X})} 1_T^X p_\Xi(X)\mu(dX) \tag{1.28}$$

Intuitively, the belief-mass functional counts the patterns, weighted with their probability of occurence, whose points fall *all* inside a subset $T \subseteq \mathcal{X}$.

The notion of PGFl is similar to the belief-mass but allows a broader range of membership functions than the indicator function [Mahl 03a]:

**Definition 1.7.** *The PGFl $G_\Xi$ of RFS $\Xi$ is the functional given by:*

$$G_\Xi[h] \overset{def}{=} \mathbb{E}[h^{\Xi(\omega)}] \tag{1.29}$$

*where* $h : \mathcal{X} \to [0\ 1]$ *is a dimensionless function and, for any function* $h$, $h^X = \prod_{x \in X} h(x)$ *(*$h^{\emptyset} = 1$ *by convention). Besides, if* $\Xi$ *admits a probability density* $p_\Xi$*:*

$$G_\Xi[h] = \int_{\mathcal{F}(\mathcal{X})} h^X p_\Xi(X) \mu(dX) \tag{1.30}$$

The function $h$ must be dimensionless for the expectation in (1.29) to be well-defined. Unless otherwise stated, the argument function of a PGFl - or, more generally, of any functional described in this thesis - is assumed to be dimensionless. Note the similarity with the *probability generating function* (PGF):

**Definition 1.8.** *The PGF* $G_X$ *of a random variable* $X$ *on* $\mathcal{X}$ *is the function given by:*

$$G_X(h) \stackrel{def}{=} \mathbb{E}[h^{X(\omega)}] \tag{1.31}$$

*where* $h \in [0\ 1]$ *is a real number.*

Comparing definitions 1.6 and 1.7 yields:

$$\beta_\Xi(T) = G_\Xi[1_T] \tag{1.32}$$

that is, the PGFl equals the belief-mass functional when the function $h$ is a subset indicator function. However, the PGFl admits "fuzzy" membership functions as arguments [Mahl 03a]. Suppose, for example, that sensor $j$ has current FOV $F^j$ such that any target $x$ inside is detected with probability $p_d^j(.)$. Further assume that a RFS $\Xi$ describes the target configuration at that current time. Then, while $\beta_\Xi(F^j)$ is the probability that all the targets are inside the FOV, $G_\Xi[p_d^j]$ is the probability that all the targets are inside the FOV *and* are detected by the sensor.

The PGFl can be written with the Janossy measures (and Janossy densities if it admits a density) as follows [Vo 08]:

**Property 1.2.** *Let* $\Xi$ *be a RFS with PGFl* $G_\Xi$ *and Janossy measures* $\{J_\Xi^{(n)}\}_{n \in \mathbb{N}}$*. Then:*

$$G_\Xi[h] = \sum_{n=0}^{\infty} \frac{1}{n!} J_\Xi^{(n)}[h, ..., h] \tag{1.33}$$

*where:*

$$J_\Xi^{(n)}[h_1, ..., h_n] \stackrel{def}{=} \int \cdots \int h_1(x_1) ... h_n(x_n) J_\Xi^{(n)}(dx_1, ..., dx_n) \tag{1.34}$$

*or, if the Janossy measures admit densities:*

$$J_\Xi^{(n)}[h_1, ..., h_n] = \int \cdots \int h_1(x_1) ... h_n(x_n) j_\Xi^{(n)}(x_1, ..., x_n) dx_1 ... dx_n \tag{1.35}$$

The proof is given in appendix A. The PGFl can be extended to the multivariate case; the following definition will be particularly useful in the extension of the PHD filter in the multi-sensor case [Vo 08]:

**Definition 1.9.** *For any $n \in [1\ N]$, let $\Xi_n : \Omega \to \mathcal{X}_n$ be a RFS with PGFl $G_{\Xi_n}$ and $h_n : \mathcal{X}_n \to [0\ 1]$. Then, the joint PGFl $G_{\Xi_1,...,\Xi_N}$ of RFSs $\Xi_1, ..., \Xi_N$ is defined by:*

$$G_{\Xi_1,...,\Xi_N}[h_1, ..., h_N] \overset{def}{=} \mathbb{E}\left[h_1^{\Xi_1(\omega)}...h_N^{\Xi_N(\omega)}\right] \tag{1.36}$$

All the notions such as probability measures and densities, functional derivatives and such are easily extended to the multivariate case. Intuitively, since each RFS $\Xi_n$ is defined on its own space $\mathcal{X}_n$, the multivariate PGFl can be easily built with the (univariate) PGFl of each RFS.

The PGFl of a union RFS can be built through the PGFls of the base RFSs [Vo 08]:

**Property 1.3.** *For any $n \in [1\ N]$, let $\Xi_n : \Omega \to \mathcal{X}$ be a RFS with PGFl $G_{\Xi_n}$. Then, if the base RFSs $\Xi_n$ are statistically independent, the PGFl of the union RFS $\bigcup_{n=1}^{N} \Xi_n$ is the product of PGFls $G_{\Xi_i}$:*

$$G_{\Xi_1 \cup ... \cup \Xi_N}[.] = \prod_{n=1}^{N} G_{\Xi_n}[.] \tag{1.37}$$

The independence between the base RFSs is important here. Intuitively, if they are independent, the event that their realization have common points has probability zero and a realization of the union RFS can be decomposed on pairwise disjoint realizations of each RFS. A sketch of the proof for $N = 2$ is given here:

$$\begin{aligned} G_{\Xi_1 \cup \Xi_2}[h] &= \sum_{n=0}^{\infty} \frac{1}{n!} J^{(n)}_{\Xi_1 \cup \Xi_2}[h, ..., h] \\ &= \sum_{n=0}^{\infty} \frac{1}{n!} \sum_{p=0}^{n} \binom{n}{p} J^{(p)}_{\Xi_1}[h, ..., h] J^{(n-p)}_{\Xi_2}[h, ..., h] \\ &= \sum_{n=0}^{\infty} \sum_{p=0}^{n} \frac{1}{p!(n-p)!} J^{(p)}_{\Xi_1}[h, ..., h] J^{(n-p)}_{\Xi_2}[h, ..., h] \end{aligned}$$

$$\begin{aligned} G_{\Xi_1\cup\Xi_2}[h] &= \sum_{p=0}^{\infty}\sum_{n=p}^{\infty}\frac{1}{p!(n-p)!}J_{\Xi_1}^{(p)}[h,...,h]J_{\Xi_2}^{(n-p)}[h,...,h] \\ &= \sum_{p=0}^{\infty}\frac{1}{p!}J_{\Xi_1}^{(p)}[h,...,h]\left(\sum_{n=p}^{\infty}\frac{1}{(n-p)!}J_{\Xi_2}^{(n-p)}[h,...,h]\right) \\ &= \left(\sum_{p=0}^{\infty}\frac{1}{p!}J_{\Xi_1}^{(p)}[h,...,h]\right)\left(\sum_{n=0}^{\infty}\frac{1}{n!}J_{\Xi_2}^{(n)}[h,...,h]\right) \\ &= G_{\Xi_1}[h]G_{\Xi_2}[h] \end{aligned}$$

One may suppose that, if RFSs $\Xi_1$ and $\Xi_2$ were not independent, the "split" of the Janossy measure of the union RFS would not have been simple and the expression of the joint PGFl would have been more complicated. In the design of the PHD filter, though, union RFS will always be built as union of independent RFSs (see chapter 2). For simplicity's sake, property 1.3 is admitted in the general case.

### Functional derivatives

The notion of functional derivative is fundamental in RFS theory. Working on PGFls and their derivatives allows the computation of probability densities, moment densities, Janossy densities and such; it is therefore possible to study the RFS from the "PGFl point of view" and deals with functions defined on the state space $\mathcal{X}$ rather than densities and measures defined on the much "larger" space $\mathcal{F}(\mathcal{X})$.

In this thesis the functional derivatives are defined and studied with the notations given by [Vo 08], but we will gradually change to the much lighter Mahler's notations [Mahl 03a], which are very easy to manipulate in the construction of the PHD filter (see chapter 2).

**Definition 1.10.** *Let $G$ be a functional and $h$, $\{g_n\}_{n\in\mathbb{N}}$ real-valued functions defined on $\mathcal{X}$. The n-th order functional derivatives, respective to $h$ and in directions $g_i$, are defined by:*

$$G^{(1)}[h;g_1] = \lim_{\epsilon\to 0^+}\frac{G[h+\epsilon g_1]-G[h]}{\epsilon_1} \tag{1.38}$$

*and, recursively:*

$$G^{(n)}[h;g_1,...,g_n] = \lim_{\epsilon\to 0^+}\frac{G^{(n-1)}[h+\epsilon g_n;g_1,...,g_{n-1}]-G^{(n-1)}[h;g_1,...,g_{n-1}]}{\epsilon} \tag{1.39}$$

*Besides, $G^{(0)}[h] = G[h]$ by convention.*

Pay attention to the fact that, unlike the argument function $h$, the directions $g_i$ *need not be* dimensionless. If $g_1$ has unit $K_{\mathcal{X}}^{-1}$, since $h+\epsilon g_1$ must be dimensionless

in (1.38), $\epsilon$ has unit $K_{\mathcal{X}}$ and thus $G^{(1)}[h; g_1]$ has unit $K_{\mathcal{X}}^{-1}$. In this thesis, unless otherwise specified, the direction functions are assumed dimensionless in order to keep the successive derivations of a PGFl dimensionless.

The next result is fundamental because it links the $n$-th order derivative of a PGFl with the $n$-th order Janossy measure and the $n$-th order factorial moment. First, a lemma is required:

**Lemma 1.1.** *Let $\Xi$ be a RFS with PGFL $G_\Xi$ and Janossy measures $\{J_\Xi^{(n)}\}_{n\in\mathbb{N}}$. Let $\{g_n\}_{n\in\mathbb{N}}$ be real-valued functions defined on $\mathcal{X}$ in [0 1]. Then:*

$$G_\Xi^{(n)}[h; g_1, ..., g_n] = \sum_{p=0}^{\infty} \frac{1}{p!} J_\Xi^{(n+p)}[g_1, ..., g_n, \underbrace{h, ...h}_{p}] \tag{1.40}$$

The proof is given in appendix A. Then follows the main result, adapted from [Vo 08]:

**Property 1.4.** *Let $\Xi$ be a RFS with PGFL $G_\Xi$, Janossy measures $\{J_\Xi^{(n)}\}_{n\in\mathbb{N}}$ and factorial moments $\{V_\Xi^{(n)}\}_{n\in\mathbb{N}}$. Let $\{g_n\}_{n\in\mathbb{N}}$ be a family of real-valued functions defined on $\mathcal{X}$ in [0 1]. Then:*

$$G_\Xi^{(n)}[0; g_1, ..., g_n] = J_\Xi^{(n)}[g_1, ..., g_n] \tag{1.41}$$

$$G_\Xi^{(n)}[1; g_1, ..., g_n] = V_\Xi^{(n)}[g_1, ..., g_n] \tag{1.42}$$

*If $\Xi$ admits a probability density, then:*

$$G_\Xi^{(n)}[0; \delta_{x_1} K_{\mathcal{X}}, ..., \delta_{x_n} K_{\mathcal{X}}] = j_\Xi^{(n)}(x_1, ..., x_n) K_{\mathcal{X}}^n \tag{1.43}$$

$$G_\Xi^{(n)}[1; \delta_{x_1} K_{\mathcal{X}}, ..., \delta_{x_n} K_{\mathcal{X}}] = v_\Xi^{(n)}(x_1, ..., x_n) K_{\mathcal{X}}^n \tag{1.44}$$

*where $\{x_n\}_{n\in\mathbb{N}}$ is a collection of points in $\mathcal{X}$ and $\delta_x(.)$ is the Dirac delta function.*

The proof is given in appendix A. Dirac functions in equations (1.43) and (1.44) are merely a practical notation allowing an easier writing of set derivations. For example, result (1.43) is easily recovered with the Delta notations. Indeed, according to (1.41):

$$\begin{aligned} G_\Xi^{(n)}[0; \delta_{y_1} K_{\mathcal{X}}, ..., \delta_{y_n} K_{\mathcal{X}}] &= J_\Xi^{(n)}[\delta_{y_1} K_{\mathcal{X}}, ..., \delta_{y_n} K_{\mathcal{X}}] \\ &= \int \cdots \int \delta_{y_1}(x_1) K_{\mathcal{X}} ... \delta_{y_n}(x_n) K_{\mathcal{X}} J_\Xi^{(n)}(dx_1, ..., dx_n) \\ &= \int \cdots \int \delta_{y_1}(x_1) ... \delta_{y_n}(x_n) j_\Xi^{(n)}(x_1, ..., x_n) K_{\mathcal{X}}^n dx_1 ... dx_n \\ &= j_\Xi^{(n)}(y_1, ..., y_n) K_{\mathcal{X}}^n \end{aligned}$$

Since the derivation in dimensionless directions is favored in this thesis and the Dirac delta function has unit $K_{\mathcal{X}}^{-1}$, units terms are explicit in equation (1.43) - unlike the equivalent form used by Vo [Vo 08]:

$$G_{\Xi}^{(n)}[0;\delta_{x_1},...,\delta_{x_n}] = j_{\Xi}^{(n)}(x_1,...,x_n)$$

In the PHD framework PGFls are derivated exlusively in the direction of Dirac delta functions, thus the following notations:

**Notation 1.1.** *For any subset* $\{x_1,...,x_n\} \subset \mathcal{X}$*:*

$$\frac{\delta^n}{\delta x_1....\delta x_n} G_{\Xi}[h] \overset{not}{=} G_{\Xi}^{(n)}[h;\delta_{x_1}K_{\mathcal{X}},...,\delta_{x_n}K_{\mathcal{X}}] \tag{1.45}$$

$$\frac{\delta}{\delta\{x_1,...,x_n\}} G_{\Xi}[h] \overset{not}{=} G_{\Xi}^{(n)}[h;\delta_{x_1}K_{\mathcal{X}},...,\delta_{x_n}K_{\mathcal{X}}] \tag{1.46}$$

That is, the $\delta$ notation will be used for the "derivation" of functionals in points from the state space $\mathcal{X}$ (or observation spaces $\mathcal{Z}^j$). The same notation will be used on multivariate functionals when there is no ambiguity on the function which is derivated. For example, if $G[g,h]$ is the functional where $g$ is defined on $\mathcal{Z}$ and $h$ is defined on $\mathcal{X}$, then $\frac{\delta^2}{\delta x \delta z} G[g,h]$ - or, equivalently, $\frac{\delta^2}{\delta z \delta x} G[g,h]$ - is the functional derivative of $G[g,h]$, respective to $g$, in direction $\delta_z(.)K_{\mathcal{Z}}$ *and*, respective to $h$, in direction $\delta_x(.)K_{\mathcal{X}}$.

The classical derivation rules being provided by the FISST calculus rules for functional derivations [Mahl 07b], the pseudo-derivation $\delta$ will be often considered as a standard derivation for calculus purposes, even though the underlying Dirac delta function induces specific properties as seen in the following paragraph. Note that equation (1.43) implies that the point order in a derivation is arbitrary - i.e. $\frac{\delta^2}{\delta x_1 \delta x_2} G_{\Xi}[h] = \frac{\delta^2}{\delta x_2 \delta x_1} G_{\Xi}[h]$ - and that any PGFl derivated twice on the same point vanishes - i.e. $\frac{\delta^2}{\delta x_1 \delta x_1} G_{\Xi}[h] = 0$.

### Calculus properties

The first property provides two useful rules for general calculus on functionals [Mahl 03a, Mahl 07b]:

**Property 1.5.** *Let $G$ be a functional, $x_0$ a fixed point in $\mathcal{X}$ and $p$ a real-valued function on $\mathcal{X}$. If, for any real-valued function $h$ defined on $\mathcal{X}$, $G[h] = h(x_0)$, then:*

$$\forall x \in \mathcal{X},\ \frac{\delta G}{\delta x}[h] = \delta_{x_0}(x) K_{\mathcal{X}} \tag{1.47}$$

*If, for any real-valued function $h$ defined on $\mathcal{X}$, $G[h] = \int_{\mathcal{X}} h(x)p(x)dx$, then:*

$$\forall x \in \mathcal{X},\ \frac{\delta G}{\delta x}[h] = p(x) K_{\mathcal{X}} \tag{1.48}$$

The proof is given in appendix A. Note that in the property 1.5 the functional $G$ may be seen as the PGFl of a very simple RFS $\Xi$ whose Janossy densities are all zero but the first order which is $j_{\Xi}^{(1)}(.) = \delta_{x_0}(.)$ in equation (1.47) and $j_{\Xi}^{(1)}(.) = p(.)$ in equation (1.48). Not surprinsingly, the argument function $h$ does not appear in any of the two results: the associated RFS having trivial cardinality distributions such that each realization has exactly one point, $h$ appears in the zero order term (in $\epsilon$) of the derivation process and is "killed" in the difference $G[h+\epsilon\delta_x] - G[h]$ (see the proof for more details). Because $h$ can be arbitrary set at 0 or 1, it shows that the notions of first order Janossy density and intensity are identical for "one-element" RFSs.

Using the factorial moments is sometimes an easy way to transform a *set integral* over on $\mathcal{F}(\mathcal{X})$ to a classical integral defined on a "smaller" space (adapted from [Mahl 03a]):

**Property 1.6.** *Let $\Xi$ be a RFS with probability density $p_{\Xi}$ and intensity $v_{\Xi}$. Then, for any $h : \mathcal{X} \to [0\ 1]$:*

$$\int_{\mathcal{F}(\mathcal{X})} h_X p_{\Xi}(X)\mu(dX) = \int_{\mathcal{X}} h(x) v_{\Xi}(x) dx \tag{1.49}$$

*where $h_X = \sum_{x \in X} h(x)$ ($h_{\emptyset} = 0$). For any point $x_0 \in \mathcal{X}$, setting $h(.) = \delta_{x_0}(.)K_{\mathcal{X}}$ gives:*

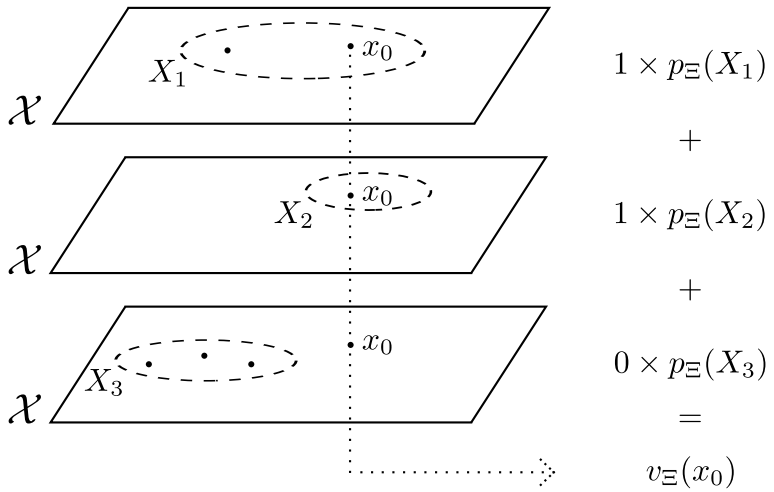
$$\int_{\mathcal{F}(\mathcal{X})} \delta_X(x_0) p_{\Xi}(X)\mu(dX) = v_{\Xi}(x_0) \tag{1.50}$$

*where $\delta_X(x_0) = \sum_{x \in X} \delta_x(x_0)$ ($\delta_{\emptyset} = 0$).*

The proof is given in appendix A. The spirit of this property is the following. Rather than averaging the sum of a function $h$ over each point of every *set*, weighted with the probability of occurence of that set, one can average $h$ over every *point*, weighted

with the intensity of this point, i.e. the probability of occurence of all sets containing this point.

**Figure 1.2** Illustration of equation (1.50)




Note that result (1.49) can be extended to higher order of factorial moments. For example, the second order equivalent would be:

$$\int_{\mathcal{F}(\mathcal{X})} h_{X,X} p_\Xi(X)\mu(dX) = \int_{\mathcal{X}^2} h(x)h(y)v_\Xi^{(2)}(x,y)dxdy \tag{1.51}$$

where $h_{X,X} = \sum_{\substack{x_i,x_j \in X \\ i<j}} h(x_i)h(x_j)$.

The last property is a "technical" result pertaining to the composition of intensities (adapted from [Mahl 03a]):

**Property 1.7.** *Let $\Phi[.]$ be a functional transformation on the real-valued functions $h : \mathcal{X} \to [0\ 1]$ such that for any $h$, $\Phi(h) : \mathcal{X} \to [0\ 1]$ and $\Phi[1] = 1$. Let $\Xi$ be a RFS with PGFl $G_\Xi$, probability density $p_\Xi$ and intensity $v_\Xi$. Assuming that there exist:*

- *a RFS $\Xi_\Phi$ with PGFl $G_{\Xi_\Phi}[.] = G_\Xi[\Phi[.]]$ and intensity $v_{\Xi_\Phi}$;*
- *for any $x \in \mathcal{X}$, a RFS $\Xi_{\Phi,x}$ with PGFl $G_{\Xi_{\Phi,x}}[.] = \Phi[.](x)$ and intensity $v_{\Xi_{\Phi,x}}$.*

*then:*

$$\forall x_0 \in \mathcal{X}, v_{\Xi_\Phi}(x_0) = \int_{\mathcal{X}} v_{\Xi_{\Phi,x}}(x_0)v_\Xi(x)dx \tag{1.52}$$

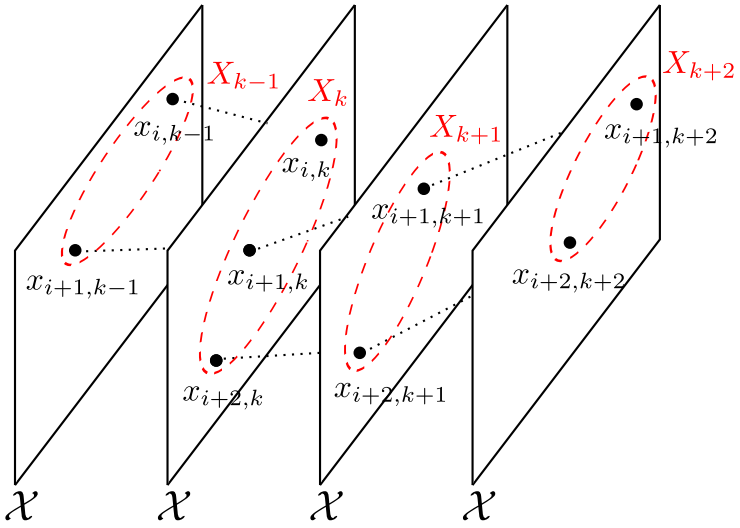
The proof is given in appendix A.

## 1.2 Multi-target filtering within the RFS framework

### 1.2.1 Principle

The RFS theory naturally applies to the modelization of a multi-target tracking problem, since there are a finite number of points - the target states - whose configuration in a space - the target state space - varies through time.

**Figure 1.3** Principle of multi-target set-based representation




Easily enough, the states $x_{i,k}$ of the true targets at time $k$ can be encapsulated in a single multi-object state $X_k \in \mathcal{F}(\mathcal{X})$ as shown in figure 1.3:

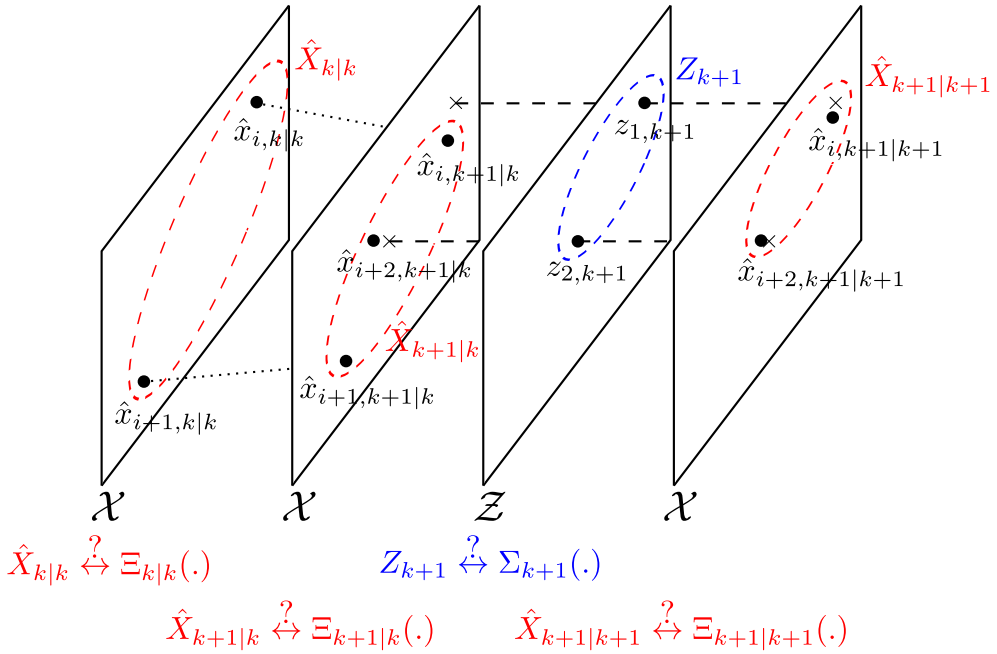
$$X_k = \bigcup_{i \in T(k)} x_{i,k} \tag{1.53}$$

where $T(k) \in \mathbb{N}$ is the label set of existing targets at time $k$ (note that the labeling of the true targets is unique, i.e. even if a target dies its label will not be used for future targets). Likewise, the measurements $z_{i,k}^j$ produced at time $k$ can be encapsulated in a single multi-objet measurement $Z_k \in \bigsqcup_{j=1}^{S} \mathcal{F}(\mathcal{Z}^j)$:

$$Z_k = \bigsqcup_{j=1}^{S} Z_k^j = \bigsqcup_{j=1}^{S} \bigcup_{i=1}^{m_k^j} z_{i,k}^j \tag{1.54}$$

where $m_k^j \in \mathbb{N}$ is the number of current measurements produced by sensor $j$ at time $k$. Note that this set-based representation is valid under the assumptions that, at any time $k$, no targets may share the same state - as already stated in section 1.1 - nor measurements from the same sensor share the same value.

**Figure 1.4** Principle of RFS filtering (time $k+1$)




As indicated by the data flow (figure 1), the RFS-based filtering process follows the classical Bayesian scheme in which one proceed with *time update* and *data update* steps sequentially (see figure 1.4). In the filtering process, there are four RFSs of interest at each time step:

**Definition 1.11.** *At any time $k+1$, there are four multi-object RFSs of interest:*

*1. The (multi-target) transition RFS $\Xi^T_{k,k+1}(X)$, with probability density $p_{\Xi^T_{k,k+1}(X)}(.)$ (or $p_{\Xi^T_{k,k+1}}(.|X)$), describes the target configuration at time $k+1$ conditionally on the (estimated) target configuration $X$ at time $k$.*

*2. The (multi-target) predicted RFS $\Xi_{k+1|k}$, with probability density $p_{\Xi_{k+1|k}}(.|Z_{1:k})$, describes the target configuration at time $k+1$ conditionally on the measurements produced up to time $k$.*

*3. The (multi-measurement) observation RFS $\Sigma_{k+1}(X)$, with probability density $p_{\Sigma_{k+1}(X)}(.)$ (or $p_{\Sigma_{k+1}}(.|X)$), describes the measurement configuration at time $k+1$ conditionally on the (estimated) target configuration $X$ at time $k+1$.*

*4. The (multi-target) posterior RFS $\Xi_{k+1|k+1}$, with probability density $p_{\Xi_{k+1|k+1}}(.|Z_{1:k+1})$, describes the target configuration at time $k+1$ conditionally on the measurements produced up to time $k+1$.*

The following proposition describes the principle of the RFS-based multi-target filter [Mahl 02]:

**Proposition 1.1.** *Assuming that:*

- *the transition densities* $\{p_{\Xi^T_{k,k+1}}\}_{k\geqslant 0}$ *are known;*
- *the observation densities* $\{p_{\Sigma_{k+1}}(.|X)\}_{k\geqslant 0}$ *are known for any set* $X$*;*
- *for any* $k \geqslant 1$*, the set of current measurements* $Z_k$ *is available at time* $k$*;*
- *an initial posterior density* $p_{\Xi_{1|0}}$ *is given.*

*then the sequences of predicted and posterior densities are given by the time update and data update equations:*

$$p_{\Xi_{k+1|k}}(.|Z_{1:k}) = \int_{\mathcal{F}(\mathcal{X})} p_{\Xi^T_{k,k+1}}(.|X) p_{\Xi_{k|k}}(X|Z_{1:k}) \mu(dX) \tag{1.55}$$

$$p_{\Xi_{k+1|k+1}}(.|Z_{1:k+1}) = \frac{p_{\Sigma_{k+1}}(Z_{k+1}|.) p_{\Xi_{k+1|k}}(.|Z_{1:k})}{\int_{\mathcal{F}(\mathcal{X})} p_{\Sigma_{k+1}}(Z_{k+1}|X) p_{\Xi_{k+1|k}}(X|Z_{1:k}) \mu(dX)} \tag{1.56}$$

This result is the extension of the well-known Bayes equations to the RFS framework. Note that the required assumptions are usual in this context. Notably, the transition densities are assumed to be known through the topography of the surveillance region, some heuristics about the typical behavior of targets, etc. Likewise, the observation densities are known through the well identified characteristics of the sensors - probability of detection, false alarm rate, statistical noise, etc.

### 1.2.2 A tractable approximation: the PHD filter

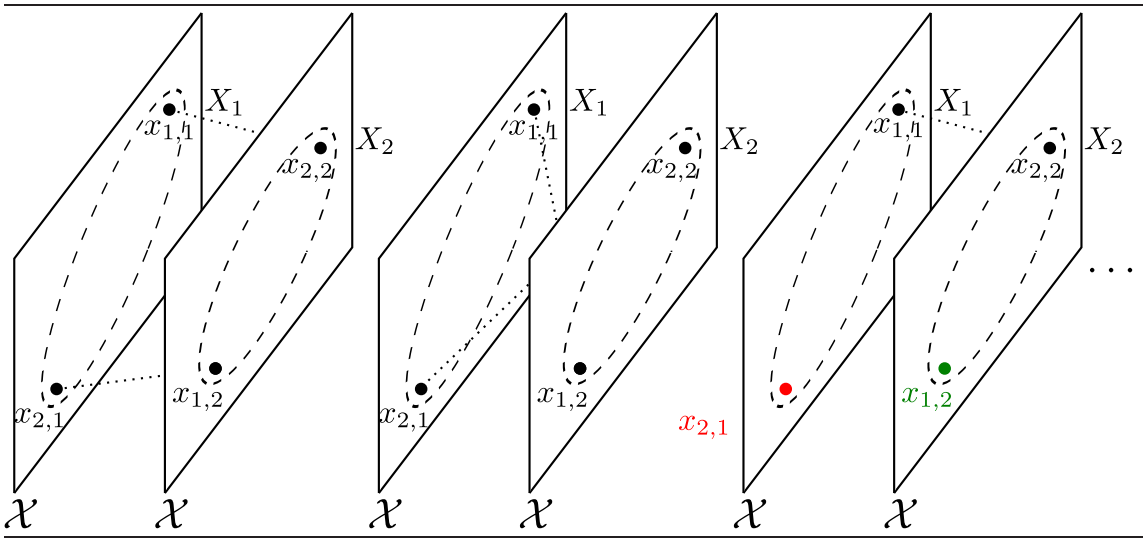
Implementing a multi-target filter based on the Bayesian recursion provided by proposition 1.1 is quite challenging and generally intractable. The main sources of untractability are:

- the design of the transition and observation densities;
- the computation of the set integrals in the update equations (1.55) and (1.56).

Recall from section 1.2.1 that the transition densities $\{p_{\Xi^T_{k,k+1}}\}_{k\leqslant 0}$ are *multi-target* densities that empnody the processes of target birth, motion, spawning - target creation conditioned on the state of an origin target, for example in the splitting of a convoy - and death. Therefore, in order to compute the probability of transition $p_{\Xi^T_{k,k+1}}(X_1|X_2)$, one must cover all the possibilities that may lead from the multi-target set $X_1$ to $X_2$. Likewise, the observation densities $\{p_{\Sigma_{k+1}}(.|X_{k+1|k})\}$ are

*multi-measurement* densities encapsulating the processes of target detection, data acquisition and false alarm, and are exceedingly difficult to design in the general case.

**Figure 1.5** Transition from multi-target set $X_1$ to $X_2$ (target birth in green, target death in red)




Additional assumptions on the target model (e.g. independence of targets) and/or the observation process (e.g. independence of data acquisition processes, maximum number of measurement per detected target, etc.) are usually necessary in order to simplify the design of the transition and observation densities. The specific assumptions required for the design of the PHD filter are fully detailed in chapter 2.

The computation of set integrals is inherently tedious, since the $\mu$ measure (1.9) covers all the possible number of elements. For example, evaluating the denominator in (1.56) requires the computation of observation $p_{\Sigma_{k+1}}(Z|.)$ and predicted $p_{\Xi_{k+1|k}}(.|Z_{1:k})$ densities for every possible multi-target set $X$, which covers any possible target number and, for a given number, any possible target states.

The main motivation behind the PHD filter is to shift the problem from the cumbersome "full space" $\mathcal{F}(\mathcal{X})$ to the "lighter space" $\mathcal{X}$ by propagating intensities $v_\Xi$ rather than the "full" densities $p_\Xi$ (adapted from [Mahl 03a]):

**Definition 1.12.** *The PHD of a RFS $\Xi$, if it exists, is its intensity $v_\Xi$.*

Equation (1.44) with the delta notation (1.45) immediately yields an expression of the PHD as a set derivative:

$$\forall x \in X,\ v_\Xi(x) = \frac{\delta G_\Xi}{\delta x}[1]K_\mathcal{X}^{-1} \tag{1.57}$$

Note that the unit term $K_{\mathcal{X}}^{-1}$ does not appear in Vo's and Mahler's work where the derivated PGFl has already unit $K_{\mathcal{X}}^{-1}$.

The PHD-based filter *in the single-sensor case only* is given by the following proposition (adapted from [Mahl 03a]):

**Proposition 1.2.** *Under the PHD filtering assumptions (tables 2.1 and 2.2), and assuming that there is only one sensor, the sequences of predicted and posterior PHDs are given by the time update and data update equations:*

$$v_{\Xi_{k+1|k}}(.|Z_{1:k}) = \int_{\mathcal{X}} \left(p^s_{k,k+1}(x) f^t_{k,k+1}(.|x) + \lambda^s_{k,k+1}(x) s_{k,k+1}(.|x)\right) v_{\Xi_{k|k}}(x|Z_{1:k})dx + \lambda^b_{k,k+1} b_{k,k+1}(.) \tag{1.58}$$

$$v_{\Xi_{k+1|k+1}}(.|Z_{1:k+1}) = \left(1 - p^d_{k+1}(.) + \sum_{z \in Z_{k+1}} \frac{p^d_{k+1}(.) L^z_{k+1}(.)}{\lambda^c_{k+1} c_{k+1}(z) + v_{\Xi_{k+1|k}}[p^d_{k+1} L^z_{k+1}]}\right) v_{\Xi_{k+1|k}}(.|Z_{1:k}) \tag{1.59}$$

*where:*

- $p^s_{k,k+1}(.)$ *is the (single-target) survival probability;*
- $f^t_{k,k+1}(.|x)$ *is the (single-target) transition density conditionally on target state* $x$*;*
- $\lambda^s_{k,k+1}(x) s_{k,k+1}(.|x)$ *is the (single-target) spawning intensity conditionally on target state* $x$*;*
- $\lambda^b_{k,k+1} b_{k,k+1}(.)$ *is the (single-target) birth intensity;*
- $p^d_{k+1}(.)$ *is the (single-target) detection probability;*
- $L^z_{k+1}(.)$ *is the (single-target) likelihood in measurement* $z$*;*
- $\lambda^c_{k+1} c_{k+1}(.)$ *is the (single-measurement) false alarm intensity;*
- $v_{\Xi_{k+1|k}}[.]$ *is the functional* $v_{\Xi_{k+1|k}}[h] \stackrel{def}{=} \int_{\mathcal{X}} h(x) v_{\Xi_{k+1|k}}(x|Z_{1:k})dx$.

All these functions will be specified in chapter 2, devoted to the rigorous construction of the PHD. Nevertheless, the equations above show the main adavantage of the PHD filter: all set-based equations and integrals have been replaced by classical equivalents. In fact, many functions appearing in equations (1.58) and (1.59) - such

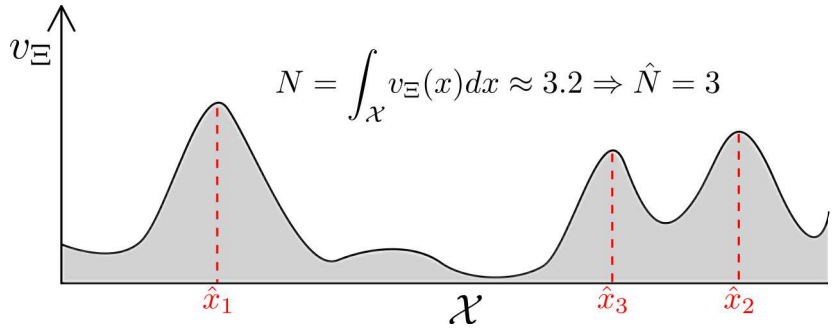
as likelihoods, probabilities of survival or single-target transition functions - are common in tracking techniques based on single-target Bayes filter. Other functions such as the birth density are more specific to the RFS framework. Besides, under the PHD filtering assumptions (see chapter 2), the multi-target RFSs $\Xi$ (predicted or posterior) are characterized by their PHDs:

**Proposition 1.3.** *Under the PHD filtering assumptions (tables 2.1 and 2.2), at any time k:*

- *the predicted (resp. posterior) estimated target number is* $N_{k+1|k} = \int_{\mathcal{X}} v_{\Xi_{k+1|k}}(x|Z_{1:k})dx$ *(resp.* $N_{k+1|k+1} = \int_{\mathcal{X}} v_{\Xi_{k+1|k+1}}(x|Z_{1:k+1})dx$*);*
- *the predicted (resp. posterior) targets are i.i.d. according to* $\frac{v_{\Xi_{k+1|k}}(.|Z_{1:k})}{N_{k+1|k}}$ *(resp.* $\frac{v_{\Xi_{k+1|k+1}}(.|Z_{1:k+1})}{N_{k+1|k+1}}$*).*

Propagating the PHD of the multi-target RFSs $\Xi$ using proposition 1.2 then extracting the information on targets using proposition 1.3 is the essence of the PHD-based filter in detection and tracking problems.

**Figure 1.6** Illustration of proposition (1.3)




Note that the PHD filter is designed for the single-sensor case *only*. Recent papers of Mahler [Mahl 09a, Mahl 09b] provide a generalization to the two-sensor case, but to the author's knowledge no extension to the multi-sensor case has been established yet.

### 1.2.3 A brief comparison of multi-target filtering techniques

The topic of this section is not a practical comparison of filtering techniques based on performance results, but rather the description of the different techniques on a theoretical level in order to help the reader grasping the philosphy behind the RFS-based filter with respect to more traditional approaches.

### Classical track-based multi-target filters

The common point in the classical track-based tracking techniques is that the filtered state is a collection of *tracks*, i.e. random objects whose behavior are as close as possible to the real targets' ones, such that each track is associated to an eventual real target. A track is usually composed of an estimation of the associated target's state, defined on the same space $\mathcal{X}$, and another object that quantifies the uncertainty regarding the estimation, usually a covariance matrix. Any track-based filter must deal with the three following issues:

- the creation of new tracks;
- the data association and the update of existing tracks;
- the deletion of existing tracks.

Arguably, the main difficulty arises in the data association process which deals with the association between living tracks and new measurements, especially in a multi-target and/or cluttered environment where the track-to-measurement step may be tedious. The methods mainly differ through their data association process.

The *multiple hypothesis tracker* (MHT) is a differed decision approach to the data association problem which focuses on updating the tracks according to the *most probable* measurement-to-track association [Blai 00, Vo 08]. Whenever a new collection of measurements is available, all measurement-to-track hypotheses are considered (with living targets as well as new ones) and associated to a probability denoting its likelihood among other hypotheses. Previous hypotheses are also considered, their probilities being updated with the Bayes rule. Thus, at each time step, a full tree of possible hypothesis is maintained, and the tracks are updated using a standard Kalman filter with the most probable association hypothesis. Keeping trace of the tree of all hypothesis allows the tracker to "change opinion" and to consider an association that was previously discarded if its associated hypothesis'probability increases. The main drawback of this method is its computational cost, which increases dramatically with the number of targets and/or the number of measurements. It can be reduced by introducing a *gating* process [Blai 00], in which associations between an estimated target state and any measurement whose distance with the target falls above a given treshold is immediately discarded. On a more practical side, the differed decision approach may be a source of "discontinuity" in the display of track states [Blai 00]: if the tracker changes the chain of previous associations due to a recent update of the associated probabilities, the track number and track states may change dramatically from one time step to the next, inducing a "discontinuous" display that may appear erratic to the operator.

The *joint probabilistic data association* (JPDA) is another, more "continuous" approach [Blai 00]. Similarly to the MHT approach, association between existing tracks and new measurements are given probabilities according to their likelihood; then, each track is updated with a Kalman update using an average of all the new measurements weighted with their association probabilities. In contrast to the MHT approach, there is no "going back" since every possible association is considered in the track update; therefore, there is no need to maintain a full tree of hypotheses. Note that some gating may be used as well in order to discard unlikely hypotheses in the averaging process. Not surprisingly, the JPDA approach is significantly lighter that the MHT [Blai 00], but has poorer performances in close targets environments because the averaging step tends to merge nearby tracks. Besides, the JPDA is designed to work with a known number of targets since each association hypothesis considers that a measurement either originates from one of the existing tracks or is a false alarm. Note also that, unlike the MHT approach which incorporates target birth and target death processes in the tree of possible hypotheses, the JPDA focuses on the update on living tracks with new measurements and requires additionnal mechanics for track creation and deletion. In its most simplified form, only the nearest measurement is taken into account in the track update and this method is known as the *nearest neighbor* (NN-JPDA or NN) approach.

### The joint multi-target probability density filter

A more recent approach, the *joint multi-target probability density* (JMPD), aims at avoiding the costly data association procedures by propagating joint multi-target densities of the form $p(X, T|Z)$, where $X = \{x_1, ..., x_T\}$ is a collection of *partitions*, $T$ is the estimated target number and each partition corresponds to an estimated target's state [Kreu 05]. This formulation seems quite close to the RFS formulation; indeed, Kreucher et al. explain in [Kreu 05] that the JMPD method can be expressed in the FISST framework. This claim was actually made made in earlier papers but was contested by Mahler [Mahl 03b]. The JMPD framework looks promising because it allows the representation of a broader range of multi-target configuration than the RFS does, since $X = \{x_1, ..., x_T\}$ is a *collection* in which several partitions may share the same state. However, the particle implementation proposed by Kreucher et al. [Kreu 04, Kreu 05] remains unclear to the author. The principle is to propagate weighted particles $X_p = \{x_{p,1}, ..., x_{p,T(p)}\}$, each particle carrying its own estimation of the target number and the target states. The number of partitions in a particle can vary in the time update step to account for the birth and/or death of the targets, the weight of each particle is updated in the data update step using an extended Bayes rule for joint densities. The problem lies in the joint estimation of the target number and the target states. The estimated target number is the partition number which is shared by the largest number of particles, their weight being taken into account. Then, each estimated target state is computed as the weighted average

of partitions associated to this target among all the particles. But, since there is not labeling among the partitions in a given particle, one must be sure that the $j$-th partition in each particle is the estimation of the same target. A K-means algorithm is proposed in order to ensure that the partitions are properly reordered before the estimation step [Kreu 05], but it is unclear if this work properly among particles with different size or among particles with the same size but with partitions representing *different* targets. For example, consider two true targets $x_1$ and $x_2$ at time $k$, two particles $X_1^k = \{x_{1,1}^k x_{1,2}^k\}$ with weight $w_1^k$ and $X_2^k = \{x_{2,1}^k x_{2,2}^k\}$ with weight $w_2^k$. Further assume that $x_{1,1}^k$ and $x_{2,1}^k$ are estimations from target $x_1$, $x_{1,2}^k$ and $x_{2,2}^k$ estimations from target $x_2$. In the time update step, $x_{1,1}^k$ evolves to $x_{1,1}^{k+1}$, $x_{1,2}^k$ evolves to $x_{1,2}^{k+1}$ and $x_{2,1}^k$ evolves to $x_{2,1}^{k+1}$ according to the target motion model, but $x_{2,2}^k$ is deleted (to account for the death of targets) and another partition $x_{2,2}^{k+1}$ is created (to account for the birth of targets). Then, according to the author's understanding of the JMPD mechanisms, the K-means algorithm is likely to keep the partition order identical in both particles and conclude that partitions $x_{1,1}^{k+1}$ and $x_{2,1}^{k+1}$ are estimations of the same target - which is correct - and partitions $x_{1,1}^{k+1}$ and $x_{2,1}^{k+1}$ as well - which is incorrect.

### Comparison with the RFS-based filter

Compared to the track-based filters and the JMPD filter (although to a lesser extent), the greatest asset of the RFS-based filter 1.1 seems to be its "completedness". Because it is a well-built extension of the Bayes rule to rigorously defined random objects, the propagation of multi-target densities with the time and data update equations (1.55) and (1.56) requires no heuristics inherent to the data association step in the MHT and/or the JPDA, or the track creation and deletion processes in the JPDA. The RFS representation allows - at least in theory - the rigorous description of complicated multi-target configurations, for example strong pairwise interaction between targets, whether in birth, motion, spawning or death processes. In other words, the RFS approach may appear more "rigorous" and "complete" than other methods since it is based on rigorously defined random objects and mathematical concepts that allows for the representation of a broad range of detection and tracking problems. Arguably, the PHD is more adapted than other methods for representing the uncertainty in the estimation of the target number. For example, if the integral of the PHD in a given subset $T \in \mathcal{X}$ yields 1.5 (see figure 1.6 for an illustration of the PHD), then it means that the presence of one *or* two targets inside $T$ is likely, thereby "enticing" the sensor manager to focus some resources in this region in order to refine the estimation of the target number.

This main asset of the RFS method is perhaps also its greatest weakness: the complexity of the RFS theory makes the equations (1.55) and (1.56) intractable,

except in special cases - e.g. if the target number is fixed. The PHD approximation (see previous section 1.2.2) enjoyed a wide popularity in the last years, notably because the filtering equations (1.58) and (1.59) can be implemented with well-known methods such as Gaussian mixtures [Vo 06] or SMC methods [Vo 05]. Moreover, the construction of the PHD filter being rigorously derived from the RFS filter as shown in the next chapter, the required assumptions in the PHD framework are clearly stated. Some of these assumptions are fairly common, as the independence of targets, others are more difficult to grasp, such as the Poisson assumption on the predicted and posterior multi-target RFS. The consequence is that it is perhaps more difficult to define the class of multi-target tracking problems for which the PHD filter is well-adapted than in the MHT or JPDA cases. A common difficulty faced with a PHD-based tracker is the unstability of the target number estimation [Erdi 05]. Another approximation of the RFS filter has been recently proposed, the *cardinalized probability hypothesis density* (CPHD) [Mahl 07a, Mahl 07c]. Some of the assumptions of the PHD are relaxed, allowing a broader range of RFSs to fit for the predicted and posterior RFS. Consequently, these multi-target RFSs cannot be characterized by their sole PHD (that is, the assumptions given in proposition 1.3 do not hold anymore), and their cardinality distributions, in addition to their PHD, must be propagated. A rigorous construction of the CPHD can be found in Vo's thesis [Vo 08] and a practical implementation in [Vo 07]. The CPHD filter propagates more information than the PHD filter at the cost of an increased computational cost, but Mahler's recent work [Mahl 10b] focuses on an more tractable approximation of the CPHD.

It should also be noted that the RFS filter (as well as the PHD or the CPHD filters), by construction, suffers some limitations that are not shared by the track-based filtering methods. Unlike track-based methods that directly provide a collection of tracks, RFS-based filters provide probability densities (RFS filter) or intensities (PHD filter) from which tracks *must be extracted*, if only for display purposes. Figure 1.6 illustrates the *highest peak* extraction method for the PHD filter. Recall from proposition 1.3 that the expected target number $N$ of the multi-target RFS is given by the integral of the PHD over the whole state space (the grey area in picture 1.6), the extracted target number $\hat{N}$ is chosen as the closest integer to $N$, and the target states are extracted at the $\hat{N}$ highest peaks of the PHD. Another method, based on the extraction of "parts" of the PHD that are worth 1 in "target weight", is given in [Tobi 08]; the extraction method used in this thesis will be based on the latter and presented in the chapter related to sensor management (see chapter 3). In any case the extraction step, although fairly independent from the filtering process, is an important part of the whole process which can shape the overall performance of the tracker.

Another limitation of the RFS-based methods, on a more fundamental level, is the absence of *track history*. In track-based methods, keeping previous filtered states naturally provides an history for each track (see black dotted lines on figure 1.4). The equivalent concept in RFS-based methods is a *set history* but, since the element order in sets is arbitrary, one *cannot* extract histories for individual tracks from the collection of previous filtered states (whether probability densities in the RFS filter or intensities in the PHD filter). Of course, track histories may be inferred from successives sets in some cases. For example, given the sets $\hat{X}_{k|k}$ and $\hat{X}_{k+1|k}$ in the situation illustrated in figure 1.4, one may safely assume that $\hat{x}_{i+2,k+1|k}$ is a newborn track and that $\hat{x}_{i+1,k+1|k}$ (resp. $\hat{x}_{i,k+1|k}$) denotes the same target as $\hat{x}_{i+1,k|k}$ (resp. $\hat{x}_{i,k|k}$) because the targets are away from each other, but retracing histories in the case of closed tracks may become increasingly complicated. One must keep in mind that, unlike track-based methods, RFS-based methods *do not* propagate track histories. Provided that the partitions could be correctly labeled in each partition, the JMPD technique should be able to propagate track histories as well, but as explained before this seems to be hardly the case. A labelisation technique adapted to the PHD is proposed in [Lin 06].

An interesting field of study is the design of hybrid filters gathering the strengths of different multi-target filtering techniques. For example, recent work of Pollard et al. [Poll 09, Poll 10] focuses on an hybrid filter combining the CPHD filter for its efficency in target detection with the MHT filter for its accuracy in target localization.

## 1.3 Performance metrics

### 1.3.1 Kullback-Leibler divergence

The *Kullback-Leibler* divergence is a measure of difference between two distributions [Aoki 11]:

$$D_{KL}(p||q) = \int p(x) \log \frac{p(x)}{q(x)} dx \tag{1.60}$$

where per convention $\log \frac{p(x)}{q(x)} = 0$ if $p(x) = q(x) = 0$ and $\log \frac{p(x)}{q(x)} = \infty$ if $p(x) > 0$ and $q(x) = 0$.

The KL divergence is closely linked to the information theory and has been used as a way to estimate an information gain prior to real observations in sensor management problems. Typically, if $q(.)$ denotes the current knowledge of the observed system, $p_1(.)$ (resp. $p_2(.)$) the estimation of the future knowledge should sensor 1 (resp. sensor 2) be used, $D_{KL}(p_1||q) \leqslant D_{KL}(p_2||q)$ could indicate that using sensor 2 is likely to be the more informative. Exploiting the KL divergence as a discriminating criteria in sensor management is not a recent idea; notably, a discretized version

appears in Kastella's work [Kast 97]. In more recent works [Kreu 05, Rist 11a], the attention seems to have shifted to its generalized version, the Rényi divergence, even though Aoki et al. [Aoki 11] argued that its proper parametrization requires a solid knowledge of its theoretical properties.

In any case, the KL divergence in this thesis is not used as a sensor management tool, but solely as an offline metric in order to check a posteriori the equality between two PHDs (see chapter 4).

### 1.3.2 OSPA distance

The *Optimal Subpattern Assignment* (OSPA) is a distance which aims at quantifying the distance between two finite sets. It was specifically created by Vo for multi-object estimation purposes as an improvement of an previous distance on finite sets, the Wasserstein metric [Hoff 04], which suffers from several inconsistencies (see Vo's thesis [Vo 08] for more details). For any $p \leqslant 1$ and $c > 0$, the OPSA distance of order $p$ with cut-off $c$ is the function defined, for any subsets $X = \{x_1, ..., x_m\}$, $Y = \{y_1, ..., y_n\}$ of $\mathcal{X}$, by:

$$\bar{d}_p^{(c)}(X,Y) = \begin{cases} 0 & (m = n = 0) \\ \left( \frac{1}{n} \left( \min_{\pi \in \Pi_n} \sum_{i=1}^{m} d^{(c)}(x_i, y_{\pi(i)})^p + c^p(n-m) \right) \right)^{1/p} & m \leqslant n \\ \bar{d}_p^{(c)}(Y,X) & m > n \end{cases} \tag{1.61}$$

where $d_{\mathcal{X}}^{(c)}(.,.) = \min(d_{\mathcal{X}}(.,.), c)$ is the distance in $\mathcal{X}$ that is cut off at $c$. The construction of the OSPA distance is quite intuitive:

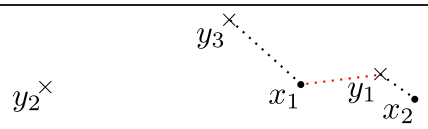
1. Match each point of the smallest set to those of the largest set so that the total distance is minimized ($\min_{\pi \in \Pi_n} \sum_{i=1}^{m} d^{(c)}(x_i, y_{\pi(i)})^p$);

2. Penalize the difference in cardinality between the two sets ($c^p(n-m)$).

In other words, the OSPA metric finds the best match between the smallest set and the same number of points in the largest set, and considers that the remaining points are "far away" - that is, their distance to any other point exceeds $c$.

**Example 1.3.** *Consider two finite subsets of* $\mathcal{X}$, $X = \{x_1, x_2\}$ *and* $Y = \{y_1, y_2, y_3\}$ *with the following configuration:*

**Figure 1.7** Example of OSPA distance




*Suppose that one is interested in the OSPA distance of order one. From the configuration of points it seems clear that:*

- $y_2$ *is the isolated point;*
- $d_{\mathcal{X}}(x_2, y_3) + d_{\mathcal{X}}(x_1, y_1) > d_{\mathcal{X}}(x_1, y_3) + d_{\mathcal{X}}(x_2, y_1)$.

*Thus, if the cut-off parameter* $c$ *is large enough:*

$$d_{\mathcal{X}}^{(c)}(x_2, y_3) + d_{\mathcal{X}}^{(c)}(x_1, y_1) > d_{\mathcal{X}}^{(c)}(x_1, y_3) + d_{\mathcal{X}}^{(c)}(x_2, y_1) \tag{1.62}$$

*That is, the OSPA matches* $x_1$ *with* $y_3$ *and* $x_2$ *with* $y_1$ *(black dotted lines on figure 1.7) and the distance is:*

$$\bar{d}_1^{(c)}(X, Y) = \frac{1}{3}\left(d_{\mathcal{X}}^{(c)}(x_1, y_3) + d_{\mathcal{X}}^{(c)}(x_2, y_1) + c\right) \tag{1.63}$$

*However, depending on the distance values, there may exist cut-off values* $d_{\mathcal{X}}(x_1, y_3) < c' < d_{\mathcal{X}}(x_2, y_3)$ *such that:*

$$c' + d_{\mathcal{X}}^{(c')}(x_1, y_1) \leqslant d_{\mathcal{X}}^{(c')}(x_1, y_3) + d_{\mathcal{X}}^{(c')}(x_2, y_1) \tag{1.64}$$

*That is, the OSPA parametrized as such matches* $x_1$ *with* $y_1$ *but considers* $x_2$ *and* $y_3$ *too far away to be matched (red dotted line on figure 1.7). The distance is then:*

$$\bar{d}_p^{(c')}(X, Y) = \frac{1}{3}\left(d_{\mathcal{X}}^{(c')}(x_1, y_1) + 2c'\right) \tag{1.65}$$

The OSPA metric is obviously appealing in the detection and tracking problems because it take into account both localization and cardinality errors, thus encapsulating the differences between, say, the set of true targets and the set of estimated targets into a single parameter. However, exploiting this metric raises several questions:

- What are the "proper values" for the order $p$ and the cut-off distance $c$?;
- How should be interpreted the value of the OSPA distance?

Vo [Vo 08] provides leads for the proper parametrization of the metric. The $p$-th order average is usual in the construction of distance and plays a similar role in the OSPA distance: as $p$ increases, $\bar{d}_p^{(c)}$ becomes increasingly sensitive to isolated points and thus penalizes more and more the "absurd" estimates. The value of the

cut-off parameter $c$ is more difficult to grasp. As explained by Vo, smaller values of $c$ emphasize on the distance between associated points, while larger values focus on the difference in the number of points. That is, the cut-off parameter $c$ balances the penalization between localization and cardinality errors. In order to keep a balanced metric, Vo advises to set the cut-off parameter significantly larger than a localization error, but significantly smaller than the maximum distance between objects. In this thesis, the typical OSPA parameters would be $p = 2$ and $c = 100$.

Because the OSPA metric aggregates all the differences between two sets in a single value, it is somewhat difficult to analyze the OSPA distance without a reference. Moreover, it can be easily shown that:

$$\forall X, \forall Y, \forall p, \qquad 0 \leqslant \bar{d}_p^{(c)}(X, Y) \leqslant c \tag{1.66}$$

That is, unlike usual metrics, the OSPA distance is capped by the cut-off parameter. Thus, as suggested by Vo, the OSPA distance might as well be evaluated with its maximum value as a reference ("the closer to the cut-off parameter $c$, the worst the estimation is") rather than its minimum value ("the closer to zero, the better the estimation is"). In any case, the author found it safer to exploit the OSPA distance solely as a comparison between different estimations and not as an objective evaluation of a single estimation.

Although the OSPA distance is a powerful metric, Vo et al. [Vo 05] argued that it has an undesirable effect when used in the PHD framework. Indeed, one must extract targets from the PHD prior to its evaluation with the OSPA (or the Wassertein) distance since a density cannot be used as input. Thus, the evaluation of the PHD through the OSPA distance depends on the target extraction process, and this dependence is generally seen as undesirable. Another limit of the OSPA distance is its inadequacy for the evaluation of labeled set (i.e. set of tracks where each track is associated to an eventual true target). That is, the OSPA will not penalize the fact that the $i$-th track is matched with the $j$-th true target at iteration $k$, but is later matched with a different true target. The extension of the OSPA so that it penalizes labelisation errors has been the focus of recent papers by Ristic et al. [Rist 10c, Rist 11b].

## 1.4 Conclusion

In this chapter, the main features of the RFS theory were presented, followed by the construction of the general RFS-based filter. This filter being usually intractable, the PHD filter, a well-known approximation, was briefly described. Designing a tracking filter within the PHD framework has some advantages compared to more traditional approaches, but the RFS framework suffers from some theoretical and

practical limitations that should be kept in mind. Finally, two useful metrics that will be applied to evaluate the performance of PHD-based trackers were described.

CHAPTER 2

# The multi-sensor PHD filter

THis chapter deals with the rigorous construction of the PHD filter as an approximation of the RFS filter, first in the single-sensor case (section 2.2), then in the multi-sensor case (section 2.3). The single-sensor case is an adaptation of Mahler's early work on the PHD [Mahl 03a, Mahl 03b] introducing the author's own notations; rewriting the single-sensor case seemed necessary in order to understand the proposed extension to the multi-sensor case, which is the main contribution of this chapter. To the author's knowledge, this is the first attempt to build the exact multi-sensor PHD filter in the general case, although the two-sensor case has already been covered by Mahler in recent works [Mahl 09a, Mahl 09b] but came later to the author's attention. Mahler's first attempts to design a tractable approximation of the multi-sensor case for practical purposes almost followed the discovery of the single-sensor case [Mahl 03b], several of these approximations will be presented in section 2.4. In a very recent paper [Liu 11], another multi-sensor extension was proposed, altough limited to linear sensor systems. Note that this chapter is in most part a clarified version of an earlier report [Dela 10].

## 2.1 Some useful RFSs

Since the RFS filter (proposition 1.1) is intractable in the general case, some assumptions on the targets and sensors must be made - that is, on the multi-object RFS involved in the RFS filter (definition 1.11 - in order to reduce to the RFS equations to tractable approximations. These definitions are adapted from [Vo 08].

### 2.1.1 Poisson RFS

**Definition 2.1.** *A Poisson RFS $\Xi$ is described by:*

1. *its cardinality distribution $\rho_\Xi(.)$, Poisson with parameter $\lambda_\Xi$: $\rho_\Xi(n) \overset{def}{=} e^{-\lambda_\Xi} \frac{\lambda_\Xi^n}{n!}$;*
2. *its spatial intensity $I_\Xi(.) \overset{def}{=} \lambda_\Xi i_\Xi(.)$, such that $\int_\mathcal{X} i_\Xi(x)dx = 1$ and the points are i.i.d. according to $i_\Xi(.)$.*

This definition is given by Vo in [Vo 08]. The Poisson RFS is one of the simplest classes of RFSs and is characterized by its spatial intensity $I_\Xi(.)$, even though it is more conveniently described with the 2-tuple $(\lambda_\Xi, I_\Xi(.))$. It accurately describes clouds of points with no particular spatial correlation between the different points. Note that the parameter $\lambda_\Xi$ is *not* required to be an integer. The following property is adapted from [Vo 08]:

**Property 2.1.** *Let $\Xi$ be a Poisson RFS with spatial intensity $I_\Xi$ and parameter $\lambda_\Xi$. Then it admits Janossy and product densities such that, for any set $X = \{x_1, ..., x_n\} \subset \mathcal{X}$:*

$$j_\Xi^{(n)}(x_1, ..., x_n) = e^{-\lambda_\Xi} \prod_{i=1}^{n} I_\Xi(x_i) \tag{2.1}$$

$$v_\Xi^{(n)}(x_1, ..., x_n) = \prod_{i=1}^{n} I_\Xi(x_i) \tag{2.2}$$

*Besides, its PGFl $G_\Xi$ is given by:*

$$G_\Xi[.] = e^{I_\Xi[.] - \lambda_\Xi} \tag{2.3}$$

*where $I_\Xi[.]$ is the functional $I_\Xi[h] \overset{def}{=} \int_\mathcal{X} h(x) I_\Xi(x)dx$.*

The proof is given in appendix A. Note that the PHD (or intensity) $v_\Xi$ of a Poisson RFS exists and equals its spatial intensity $I_\Xi$, that is, a Poisson RFS is *completely* described by its PHD.

### 2.1.2 Independent Identically Distributed Cluster RFS

**Definition 2.2.** *A (i.i.d) cluster RFS $\Xi$ is described by:*

1. *its cardinality distribution $\rho_\Xi(.)$ with mean $\lambda_\Xi$: $\lambda_\Xi \overset{def}{=} \sum_{n=0}^{\infty} n\rho_\Xi(n)$;*
2. *its spatial intensity $I_\Xi \overset{def}{=} \lambda_\Xi i_\Xi(.)$, such that $\int_\mathcal{X} i_\Xi(x)dx = 1$ and the points are i.i.d. according to $i_\Xi(.)$.*

This definition is given by Vo in [Vo 08]. The cluster RFS is a generalization of the Poisson RFS which allows a broader range of cardinality distributions. It is characterized by its spatial intensity $I_\Xi(.)$ *and* its cardinality distribution $\rho_\Xi(.)$, even though it is more conveniently described by the 3-tuple $(\rho_\Xi(.), \lambda_\Xi, I_\Xi(.))$. It is well-adapted to the description of a false alarm process, since its spatial distribution is "evenly distributed" as the Poisson RFS, but allows a greater flexibility on the number of false alarms per scan. The following property is adapted from [Vo 08]:

**Property 2.2.** *Let $\Xi$ be a cluster RFS with cardinality distribution $\rho_\Xi(.)$, mean $\lambda_\Xi$ and spatial intensity $I_\Xi$. Then it admits Janossy densities such that, for any set $X = \{x_1, ..., x_n\} \subset \mathcal{X}$:*

$$j_\Xi^{(n)}(x_1, ..., x_n) = \frac{n!\rho_\Xi(n)}{\lambda_\Xi^n} \prod_{i=1}^{n} I_\Xi(x_i) \tag{2.4}$$

*Besides, its PGFl $G_\Xi$ is given by:*

$$G_\Xi[.] = G_{|\Xi|}\left(\frac{I_\Xi[.]}{\lambda_\Xi}\right) \tag{2.5}$$

*where $I_\Xi[.]$ is the functional $I_\Xi[h] \overset{def}{=} \int_\mathcal{X} h(x) I_\Xi(x) dx$, and $G_{|\Xi|}$ is the PGF of random variable $|\Xi| : \omega \mapsto |\Xi(\omega)|$.*

The proof is given in appendix A. Note that, unlike the Poisson RFS (proposition 2.1), there is no easy expression of the factorial moments, and in the general case a cluster RFS is *not* completely described by its PHD.

### 2.1.3 Bernoulli RFS

**Definition 2.3.** *A Bernoulli RFS $\Xi$ is characterized by:*

1. *its cardinality distribution $\rho_\Xi(.)$ with parameter $b_\Xi$ such that $\rho_\Xi(0) = 1 - b_\Xi$, $\rho_\Xi(1) = b_\Xi$, $\rho_\Xi(n) = 0$ otherwise;*
2. *its spatial distribution $i_\Xi(.)$, such that the eventual point is distributed according to $i_\Xi(.)$.*

This definition is given by Vo in [Vo 08]. The Bernoulli RFS is completely described by its spatial distribution $i_\Xi(.)$ *and* its parameter $b_\Xi$. Clearly, it is well adapted to the modelization of the evolution of a target with state $x$: either it dies with probability $1 - b_{\Xi(x)}$ or it moves to a new state according to probability distribution $i_{\Xi(x)}(.)$. Likewise, it naturally describes a single-sensor/single-target detection process: a target in $x$ is either undetected with probability $1 - b_{\Xi(x)}$, or it is detected and the sensor produces a new measurement according to probability distribution $i_{\Xi(x)}(.)$. Note that it can be seen as a particular case of cluster RFS. The following property is adapted from [Vo 08]:

**Property 2.3.** *Let $\Xi$ be a Bernoulli RFS with spatial distribution $i_\Xi$ and parameter $b_\Xi$. Then it admits Janossy densities such that, for any set $X = \{x_1, ..., x_n\} \subset \mathcal{X}$:*

$$j_\Xi^{(n)}(x_1, ..., x_n) = \begin{cases} 1 - b_\Xi & n = 0 \\ b_\Xi i_\Xi(x_1) & n = 1 \\ 0 & otherwise \end{cases} \tag{2.6}$$

*Besides, its PGFl $G_\Xi$ is given by:*

$$G_\Xi[.] = 1 - b_\Xi + b_\Xi i_\Xi[.] \tag{2.7}$$

*where $i_\Xi[.]$ is the functional $i_\Xi[h] \stackrel{def}{=} \int_\mathcal{X} h(x) i_\Xi(x) dx$.*

The proof is given in appendix A. Because the Bernoulli RFSs are designed for the description of single-object behaviors (either new target or new measurement), it is interesting to define the notion of multi-Bernoulli RFS [Vo 08]:

**Definition 2.4.** *A multi-Bernoulli RFS is the union RFS of statistically independent Bernoulli RFSs .*

A multi-Bernoulli RFS is naturally characterized by the spatial intensities and parameters of the Bernoulli RFS it is built upon and its PGFl can be written as follows (adapted from [Vo 08]):

**Property 2.4.** *Let $\{\Xi_n\}_{n\in[1\ N]}$ be a family of independent Bernoulli RFS with spatial distributions $\{i_{\Xi_n}\}_{n\in[1\ N]}$ and parameters $\{b_{\Xi_n}\}_{n\in[1\ N]}$, and $\Xi$ the resulting multi-Bernoulli RFS. Then, its PGFl $G_\Xi$ is given by:*

$$G_\Xi[.] = \prod_{n=1}^{N} (1 - b_{\Xi_n} + b_{\Xi_n} i_{\Xi_n}[.]) \tag{2.8}$$

*where $i_{\Xi_n}[.]$ is the functional $i_{\Xi_n}[h] \stackrel{def}{=} \int_\mathcal{X} h(x) i_{\Xi_n}(x) dx$.*

The proof is straightforward using property 1.3.

## 2.2 Single-sensor PHD filter

### 2.2.1 Time update equation

The challenge is to find a tractable form of the Bayes time update equation (1.55):

$$p_{\Xi_{k+1|k}}(.|Z_{1:k}) = \int_{\mathcal{F}(\mathcal{X})} p_{\Xi^T_{k,k+1}}(.|X) p_{\Xi_{k|k}}(X|Z_{1:k}) \mu(dX) \tag{2.9}$$

where:

- $\Xi_{k|k}$ is the posterior (multi-target) RFS at time $k$;
- $\Xi^{T}_{k,k+1}$ is the transition (multi-target) RFS at time $k+1$;
- $\Xi_{k+1|k}$ is the predicted (multi-target) RFS at time $k+1$;

In this general form, the transition RFS may cover a broad range of targets' behaviors; however the PHD construction requires some restrictions through the following assumptions (adatpted from [Mahl 03a]):

**Proposition 2.1.** *Assuming that, at time k:*

- *a living target with state $x_{i,k}$ dies with probability $1 - p^{s}_{k,k+1}(x_{i,k})$;*
- *a surviving target with state $x_{i,k}$ evolves according to probability distribution $f^{t}_{k,k+1}(.|x_{i,k})$;*
- *from a living target with state $x_{i,k}$, a set of spawned targets $X^{S}_{i,k+1}$ is born according to probability distribution $p^{S}_{k,k+1}(.|x_{i,k})$;*
- *a set of targets $X^{B}_{k+1}$ is born spontaneously according to probability distribution $p^{B}_{k,k+1}(.)$;*
- *the evolution, spawning and birth processes are statistically independent conditionally on the set $X_k$ of living targets.*

*then the transition RFS $\Xi^{T}_{k,k+1}$ is the union RFS:*

$$\Xi^{T}_{k,k+1}(X) \overset{def}{=} \underbrace{\left(\bigcup_{x\in X} \Xi^{E}_{k,k+1}(x)\right)}_{\Xi^{E}_{k,k+1}(X)} \cup \underbrace{\left(\bigcup_{x\in X} \Xi^{S}_{k,k+1}(x)\right)}_{\Xi^{S}_{k,k+1}(X)} \cup \Xi^{B}_{k,k+1} \tag{2.10}$$

*where:*

- *$\Xi^{E}_{k,k+1}(x)$ is the (single-target) evolution Bernoulli RFS in state $x$ with spatial distribution $f^{t}_{k,k+1}(.|x)$ and parameter $p^{s}_{k,k+1}(x)$;*
- *$\Xi^{E}_{k,k+1}(X) = \bigcup_{x\in X} \Xi^{E}_{k,k+1}(x)$ is the (multi-target) evolution Multi-Bernoulli RFS in set $X$;*
- *$\Xi^{S}_{k,k+1}(x)$ is the (single-target) spawning RFS in state $x$ with probability distribution $p^{S}_{k,k+1}(.|x)$;*
- *$\Xi^{S}_{k,k+1}(X) = \bigcup_{x\in X} \Xi^{S}_{k,k+1}(x)$ is the (multi-target) spawning RFS in set $X$;*
- *$\Xi^{B}_{k,k+1}$ is the (multi-target) birth RFS with probability distribution $p^{B}_{k,k+1}(.)$.*

*Besides, the PGFl of the transition RFS* $\Xi^T_{k,k+1}$ *is given by:*

$$G_{\Xi^T_{k,k+1}(X)}[h] = (1 - p^s_{k,k+1}(.) + p^s_{k,k+1}(.) f^t_{k,k+1}[h|.])^X (G_{\Xi^S_{k,k+1}(.)}[h])^X G_{\Xi^B_{k,k+1}}[h] \quad (2.11)$$

*where:*

- $f^t_{k,k+1}[h|x]$ *is the functional* $f^t_{k,k+1}[h|x] \stackrel{def}{=} \int_{\mathcal{X}} h(y) f^t_{k,k+1}(y|x)dy$*;*
- $G_{\Xi^S_{k,k+1}(x)}$ *is the PGFl of the spawning RFS* $\Xi^S_{k,k+1}(x)$*;*
- $G_{\Xi^B_{k,k+1}}$ *is the PGFl of the birth RFS* $\Xi^B_{k,k+1}$*.*

The proof is given in appendix A. Note that no assumption on the spawning and birth models are required so far but their mutual independence. Pay attention to the fact that probability distributions $p^S_{k,k+1}(.|x)$ and $p^B_{k,k+1}(.)$ are dimensionless and *set-based* (i.e. defined on $\mathcal{F}(\mathcal{X})$), while $f^t_{k,k+1}(.|x)$ has unit $K^{-1}_{\mathcal{X}}$ and is *state-based* (i.e. defined on $\mathcal{X}$).

**Figure 2.1** Example of transition RFS (evolution in red, spontaneous birth in green, spawning in blue)

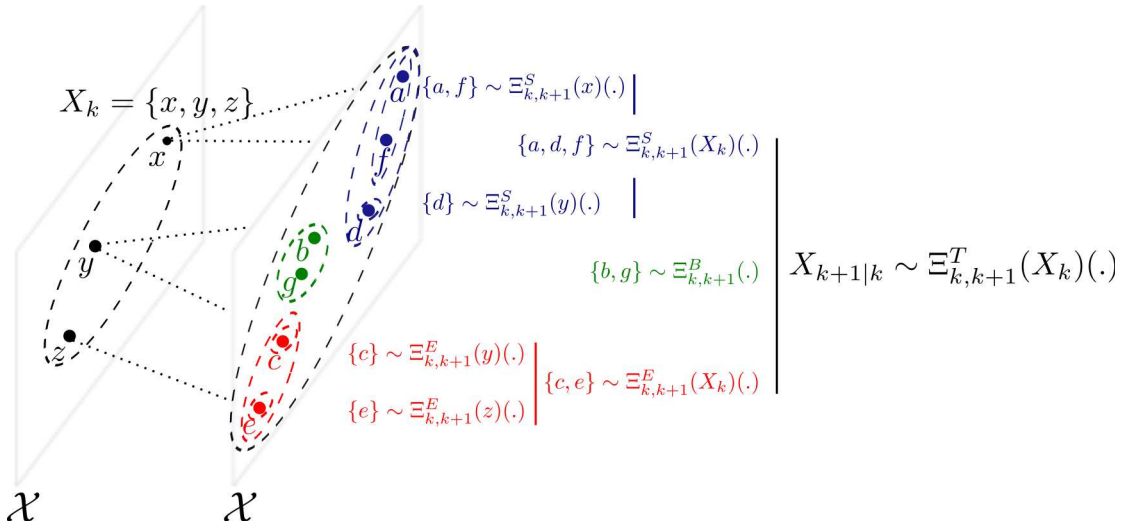



Figure 2.1 shows an example of transition RFS based on a given set $X_k$. Note that target $x$ dies but spawns two new targets (e.g. a convoy splits in two parts), target $y$ evolves and spawns a new target (e.g. a target leaves a convoy), while target $z$ evolves without spawning any new target. Besides, two targets are born independently of $x$, $y$ or $z$.

Once the transition RFS is explicitly stated, the PHD equivalent of the time update equation (2.9) can be built as follows (adatpted from [Mahl 03a]):

**Theorem 2.1.** *Under assumptions given in proposition 2.1, the time update equation of the PHD filter is given by:*

$$\begin{aligned} & v_{\Xi_{k+1|k}}(.|Z_{1:k}) \\ & = \int_{\mathcal{X}} \left( p^s_{k,k+1}(x) f^t_{k,k+1}(.|x) + v_{\Xi^S_{k+1|k}(x)}(.) \right) v_{\Xi_{k|k}}(x|Z_{1:k}) dx + v_{\Xi^B_{k+1|k}}(.) \end{aligned} \tag{2.12}$$

*Besides, the estimated target number $N_{k+1|k}$ is given by:*

$$\begin{aligned} & N_{k+1|k} \\ & = \int_{\mathcal{X}} \left[ \left( p^s_{k,k+1}(x) + \int_{\mathcal{X}} v_{\Xi^S_{k+1|k}(x)}(y) dy \right) v_{\Xi_{k|k}}(x|Z_{1:k}) \right] dx + \int_{\mathcal{X}} v_{\Xi^S_{k+1|k}(x)}(x) dx \end{aligned} \tag{2.13}$$

The proof is given in appendix A. Since no assumptions on the spawning and birth models are made, equations (2.12) and (2.13) are still intractable if PHD $v_{\Xi^S_{k+1|k}(x)}(.)$ and $v_{\Xi^B_{k+1|k}}(.)$ cannot be constructed explicitly. It is therefore common practice to consider the spawning and birth RFSs to be Poisson, which gives the much more useful result (adatpted from [Mahl 03a]):

**Corollary 2.1.** *Under the same assumptions as theorem 2.1 and the additionnal assumptions that:*

- *spawning RFSs $\Xi^S_{k,k+1}(x)$ are Poisson with intensities $\lambda^s_{k,k+1}(x) s_{k,k+1}(.|x)$;*
- *the birth RFS $\Xi^B_{k,k+1}$ is Poisson with intensity $\lambda^b_{k,k+1} b_{k,k+1}(.)$.*

*the time update equation of the PHD filter is given by:*

$$\begin{aligned} & v_{\Xi_{k+1|k}}(.|Z_{1:k}) \\ & = \int_{\mathcal{X}} \left( p^s_{k,k+1}(x) f^t_{k,k+1}(.|x) + \lambda^s_{k,k+1}(x) s_{k,k+1}(.|x) \right) v_{\Xi_{k|k}}(x|Z_{1:k}) dx + \lambda^b_{k,k+1} b_{k,k+1}(.) \end{aligned} \tag{2.14}$$

*Besides, the predicted target number $N_{k+1|k}$ is given by:*

$$\begin{aligned} & N_{k+1|k} \\ & = \int_{\mathcal{X}} \left( p^s_{k,k+1}(x) + \lambda^s_{k,k+1}(x) \right) v_{\Xi_{k|k}}(x|Z_{1:k}) dx + \lambda^b_{k,k+1} \end{aligned} \tag{2.15}$$

Table 2.1: PHD filter: assumptions for the time update equation

| Object modelization | RFS construction |
|---|---|
| Evolution process | |
| Evolution of target $x$: | $\Xi^E_{k,k+1}(x)$: Bernoulli with: |
| 1. dies with prob. $1 - p^s_{k,k+1}(x)$; | 1. parameter $p^s_{k,k+1}(x)$; |
| 2. if alive, moves acc. to $f^t_{k,k+1}(.\|x)$. | 2. spatial distribution $f^t_{k,k+1}(.\|x)$. |
| Single-target evolutions: independent | $\Xi^E_{k,k+1}(X)$: multi-Bernoulli |
| Spawning process | |
| Spawning from target $x$: | $\Xi^S_{k,k+1}(x)$: Poisson with: |
| 1. target #: Poisson, param. $\lambda^s_{k,k+1}(x)$; | 1. parameter $\lambda^s_{k,k+1}(x)$; |
| 2. targets i.i.d. acc. to $s_{k,k+1}(.\|x)$. | 2. spatial intensity $\lambda^s_{k,k+1}(x)s_{k,k+1}(.\|x)$. |
| Single-target spawnings: independent | $\Xi^S_{k,k+1}(X)$: union of independent RFSs |
| Spontaneous birth process | |
| Spontaneous birth: | $\Xi^B_{k,k+1}$: Poisson with: |
| 1. target #: Poisson, param. $\lambda^b_{k,k+1}$; | 1. parameter $\lambda^b_{k,k+1}$; |
| 2. targets i.i.d. acc. to $b_{k,k+1}(.)$. | 2. spatial intensity $\lambda^b_{k,k+1}s_{k,k+1}(.)$. |
| Transition process | |
| Evolution, spawning and birth: independent | $\Xi^T_{k,k+1}(X)$: union of independent RFSs |

### 2.2.2 Data update equation

#### Construction

The challenge is to find a tractable form of the Bayes data update equation (1.56) in the single-sensor case:

$$p_{\Xi_{k+1|k+1}}(.|Z_{1:k+1}) = \frac{p_{\Sigma_{k+1}}(Z_{k+1}|.)p_{\Xi_{k+1|k}}(.|Z_{1:k})}{\int_{\mathcal{F}(\mathcal{X})} p_{\Sigma_{k+1}}(Z_{k+1}|X)p_{\Xi_{k+1|k}}(X|Z_{1:k})\mu(dX)} \tag{2.16}$$

where:

- $\Xi_{k+1|k}$ is the predicted (multi-target) RFS at time $k+1$;
- $\Sigma_{k+1}$ is the observation (multi-measurement) RFS at time $k+1$;
- $\Xi_{k+1|k+1}$ is the posterior (multi-target) RFS at time $k+1$;

- $Z_{k+1}$ is the multi-measurement set produced by the *only* sensor available at time $k+1$.

In this general form, the observation RFS may cover a broad range of sensors' behaviors; however the PHD construction requires some restrictions through the following assumptions (adatpted from [Mahl 03a]):

**Proposition 2.2.** *Assuming that, at time $k+1$:*

- *a living target with state $x_{i,k+1}$ is detected with probability $p^d_{k+1}(x_{i,k+1})$;*
- *a detected target with state $x_{i,k+1}$ is the origin of a single true measurement according to probability distribution $f^o_{k+1}(.|x_{i,k+1}) = L_{k+1}(x_{i,k+1})$;*
- *a set of false alarm measurements (or false alarms) $Z^C_{k+1}$ are created according to a false alarm (or clutter) process, assumed Poisson with parameter $\lambda^c_{k+1}$ and intensity $\lambda^c_{k+1}c_{k+1}(.)$;*
- *the detection and false alarm processes are statistically independent conditionally on the set $X_{k+1}$ of living targets.*

*then the observation RFS $\Sigma_{k+1}$ is the union RFS:*

$$\Sigma_{k+1}(X) \stackrel{def}{=} \underbrace{\left(\bigcup_{x\in X} \Sigma^D_{k+1}(x)\right)}_{\Sigma^D_{k+1}(X)} \cup \Sigma^C_{k+1} \tag{2.17}$$

*where:*

- *$\Sigma^D_{k+1}(x)$ is the (single-measurement) detection Bernoulli RFS in state $x$ with spatial distribution $f^o_{k+1}(.|x)$ and parameter $p^d_{k+1}(x)$;*
- *$\Sigma^D_{k+1}(X) = \bigcup_{x\in X} \Sigma^D_{k+1}(x)$ is the (multi-measurement) detection Multi-Bernoulli RFS in set $X$;*
- *$\Sigma^C_{k+1}$ is the (multi-measurement) false alarm Poisson RFS with parameter $\lambda^c_{k+1}$ and intensity $\lambda^c_{k+1}c_{k+1}(.)$.*

*Besides, the PGFl of the observation RFS $\Sigma_{k+1}$ is given by:*

$$G_{\Sigma_{k+1}(X)}[g] = (1 - p^d_{k+1}(.) + p^d_{k+1}(.)f^o_{k+1}[g|.])^X e^{\lambda^c_{k+1}c_{k+1}[g]-\lambda^c_{k+1}} \tag{2.18}$$

*where:*

- *$f^o_{k+1}[g|x]$ is the functional $f^o_{k+1}[g|x] \stackrel{def}{=} \int_{\mathcal{Z}} g(z)f^o_{k+1}(z|x)dz$;*

- $c_{k+1}[g]$ *is the functional* $c_{k+1}[g] \overset{def}{=} \int_{\mathcal{Z}} g(z)c_{k+1}(z)dz$.

The proof is given in appendix A. Pay attention to the fact that spatial distribution $f^o_{k+1}(.|x)$ and intensity $\lambda^c_{k+1}c_{k+1}(.)$ have unit $K_{\mathcal{Z}}^{-1}$ and are *state-based* (i.e. defined on $\mathcal{Z}$).

**Figure 2.2** Example of observation RFS (true measurements in green, false alarms in red)

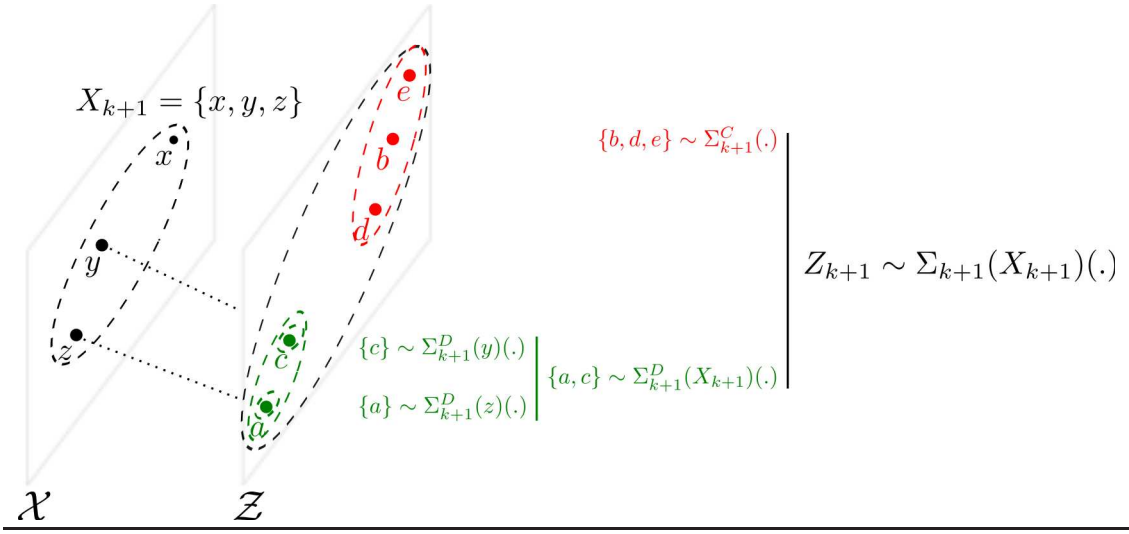



Figure 2.2 shows an example of observation RFS based on a given set $X_{k+1}$. Note that targets $y$ and $z$ are detected and the origin of one true measurement each, while target $x$ remains undetected. Besides, three false alarms are produced independently of $x$, $y$ and $z$.

The next step is the definition of *cross-term functionals* (or cross-terms) $\beta[.]$, whose extension to the multi-sensor case will be most useful for the construction of data update equation. Nonetheless, they are introduced here since they provide an intuitive representation of the data update equation in the single-sensor as well as the multi-sensor case. The notion of cross-term was introduced by the author in [Dela 10].

**Definition 2.5.** *Under the same assumptions as proposition 2.2, the single-sensor cross-term* $\beta_{k+1|k}$ *is the functional defined, for any real-valued functions* $h$ *(resp.* $g$*) defined on* $\mathcal{X}$ *(resp.* $\mathcal{Z}$*) in* $[0\ 1]$*, by:*

$$\beta_{k+1|k}[g,h] \overset{def}{=} \lambda^c_{k+1}c_{k+1}[g] - \lambda^c_{k+1} + v_{\Xi_{k+1|k}}\left[h(1 - p^d_{k+1} + p^d_{k+1}f^o_{k+1}[g|.])\right] - v_{\Xi_{k+1|k}}[1] \tag{2.19}$$

*where* $v_{\Xi_{k+1|k}}[h]$ *is the functional* $v_{\Xi_{k+1|k}}[h] \overset{def}{=} \int_{\mathcal{X}} h(x)v_{\Xi_{k+1|k}}(x|Z_{1:k})dx$.

For simplicity's sake, the time subscripts on cross-terms will be omitted when there is no ambiguity. The cross-term is a joint functional whose functional derivative (see definition 1.10) can be computed in functions defined on $\mathcal{X}$ or $\mathcal{Z}$. The following notations will be used from now on:

**Notation 2.1.** *For any $x \in \mathcal{X}$ and $z \in \mathcal{Z}$:*

$$\beta[., \delta_x] \overset{not}{=} \frac{\delta}{\delta x}\beta[., h] \tag{2.20}$$

$$\beta[\delta_z, .] \overset{not}{=} \frac{\delta}{\delta z}\beta[g, .] \tag{2.21}$$

The derivated cross-terms can be expressed as follows:

**Proposition 2.3.** *For any $x \in \mathcal{X}$ and $z \in \mathcal{Z}$:*

$$\beta[g, \delta_x] = (1 - p^d_{k+1}(x) + p^d_{k+1}(x) f^o_{k+1}[g|x]) v_{\Xi_{k+1|k}}(x|Z_{1:k}) K_{\mathcal{X}} \tag{2.22}$$

$$\beta[\delta_z, h] = \lambda^c_{k+1} c_{k+1}(z) K_{\mathcal{Z}} + v_{\Xi_{k+1|k}}[h p^d_{k+1} L^z_{k+1}] K_{\mathcal{Z}} \tag{2.23}$$

$$\beta[\delta_z, \delta_x] = p^d_{k+1}(x) L^z_{k+1}(x) v_{\Xi_{k+1|k}}(x|Z_{1:k}) K_{\mathcal{X}} K_{\mathcal{Z}} \tag{2.24}$$

*Besides, setting $g = 0, h = 1$ gives:*

$$\beta[0, \delta_x] = (1 - p^d_{k+1}(x)) v_{\Xi_{k+1|k}}(x|Z_{1:k}) K_{\mathcal{X}} \tag{2.25}$$

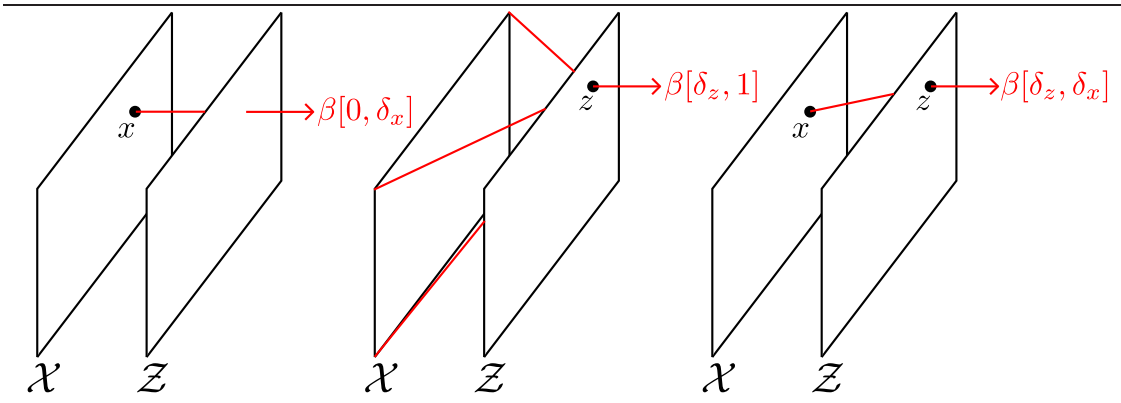
$$\beta[\delta_z, 1] = \lambda^c_{k+1} c_{k+1}(z) K_{\mathcal{Z}} + v_{\Xi_{k+1|k}}[p^d_{k+1} L^z_{k+1}] K_{\mathcal{Z}} \tag{2.26}$$

The proof is given in appendix A. As for the PGFl, the functions $g$ (resp. $h$) can be seen as a "fuzzy" membership function on measurement space $\mathcal{Z}$ (resp. target space $\mathcal{X}$). Derivated cross-terms can therefore be seen as "likelihoods", conditonnally on the predicted PHD $v_{\Xi_{k+1|k}}$, that:

- $\beta[0, \delta_x]$: a target is in state $x$ *and* is undetected;
- $\beta[\delta_z, 1]$: a measurement, whose origin is unknown, is produced in point $z$;
- $\beta[\delta_z, \delta_x]$: a target is in state $x$, is detected *and* the origin of true measurement $z$.

The term "likelihood" in the intepretation of the derivated cross-terms is an abuse of notation; more accurately they weight events associating points in target and observation spaces based on the information known on the system so far, hence their name (see figure 2.3 for an illustration).

**Figure 2.3** Illustration of the single-sensor cross-term




a) $x$ is undetected b) $z$ stems from an unkown origin c) $z$ stems from $x$

Note that $\beta[\delta_z, 1]$ covers both possible origins for measurement $z$, either a false alarm - $\lambda^c_{k+1} c_{k+1}(z) K_{\mathcal{Z}}$ - *or* a true measurement - $v_{\Xi_{k+1|k}}[p^d_{k+1} L^z_{k+1}] K_{\mathcal{Z}}$, which can be seen as an "expectation" over target space $\mathcal{X}$ considering every possible state $x$ as the potential origin of $z$. Note also that a cross-term derivated in *several* measurements (e.g. $\frac{\delta^2}{\delta z_2 \delta z_1}\beta[g,h] = \frac{\delta}{\delta z_2}\beta[\delta_{z_1}, h]$) and/or in *several* targets (e.g. $\frac{\delta^2}{\delta x_2 \delta x_1}\beta[g,h] = \frac{\delta}{\delta x_2}\beta[g, \delta_{x_1}]$) vanishes. This is consistent with the observation model (proposition 2.2) since a *single* true measurement cannot stem from *several* targets, nor can *several* true measurements stem from a *single* target.

Once the observation RFS is explicitly stated and the cross-term properly defined, the PHD equivalent of the date update equation (2.16) - in the single-sensor case only - can be built as follows:

**Theorem 2.2.** *Under the assumptions given in proposition 2.2 and the additional assumption that the predicted RFS $\Xi_{k+1|k}$ is Poisson, the data update equation of the single-sensor PHD filter is given by:*

$$v_{\Xi_{k+1|k+1}}(.|Z_{1:k+1}) = \frac{\left[\frac{\delta}{\delta x}\left(\frac{\delta}{\delta Z_{k+1}} e^{\beta[g,h]}\right)\right]_{g=0,h=1}}{\left[\frac{\delta}{\delta Z_{k+1}} e^{\beta[g,h]}\right]_{g=0,h=1}} K_{\mathcal{X}}^{-1} \tag{2.27}$$

$$= \left(\beta[\delta_\emptyset, \delta_x] + \sum_{z \in Z_{k+1}} \frac{\beta[\delta_z, \delta_x]}{\beta[\delta_z, 1]}\right) K_{\mathcal{X}}^{-1} \tag{2.28}$$

$$= \left(1 - p^d_{k+1}(x) + \sum_{z \in Z_{k+1}} \frac{p^d_{k+1}(x) L^z_{k+1}(x)}{\lambda^c_{k+1} c_{k+1}(z) + v_{\Xi_{k+1|k}}[p^d_{k+1} L^z_{k+1}]}\right) v_{\Xi_{k+1|k}}(.|Z_{1:k}) \tag{2.29}$$

*where $Z_{k+1}$ is the set of current measurements.*

The proof is given in appendix A. This last result (2.29) is the well-known tractable expression of the single-sensor PHD data equation [Mahl 03a, Mahl 03b].

Table 2.2: PHD filter: assumptions for the data update equation (single-sensor)

| Object modelization | RFS construction |
|---|---|
| Detection process | |
| Observation of target $x$:<br>1. detected with prob. $p^d_{k+1}(x)$;<br>2. if detected, meas. acc. to $f^o_{k+1}(.\|x)$.<br>Single-target observations: independent | $\Sigma^D_{k+1}(x)$: Bernoulli with:<br>1. parameter $p^d_{k+1}(x)$;<br>2. spatial distribution $f^o_{k+1}(.\|x)$.<br>$\Sigma^D_{k+1}(X)$: multi-Bernoulli |
| False alarm process | |
| False alarm:<br>1. measurement #: Poisson, param. $\lambda^c_{k+1}$;<br>2. measurements i.i.d. acc. to $c_{k+1}(.)$. | $\Sigma^C_{k+1}$: Poisson with:<br>1. parameter $\lambda^c_{k+1}$;<br>2. spatial intensity $\lambda^c_{k+1}c_{k+1}(.)$. |
| Observation process | |
| Detection and false alarm: independent | $\Sigma_{k+1}(X)$: union of independent RFSs |
| Target model | |
| Predicted configuration:<br>1. target #: Poisson, param. $v_{\Xi_{k+1\|k}}[1]$;<br>2. targets i.i.d. acc. to $\frac{v_{\Xi_{k+1\|k}}(.)}{v_{\Xi_{k+1\|k}}[1]}$. | $\Xi_{k+1\|k}$: Poisson with:<br>1. parameter $v_{\Xi_{k+1\|k}}[1]$;<br>2. spatial intensity $v_{\Xi_{k+1\|k}}(.)$. |

### Qualitative analysis

The key to theorem 2.2 is the critical assumption that the predicted RFS $\Xi_{k+1|k}$ is Poisson, which greatly simplifies the derivation of the PGFl (see the proof for more details). It is somewhat difficult to evaluate its validity in practical detection and tracking problems, because the Poisson characterization of an RFS is not easily linked to single-object behavioral patterns. However, pay attention to the fact that the principle of RFS filtering is *not* to find a *static* multi-target RFS whose sequential realizations matches best the successive target configurations, but rather to modify *dynamically* a multi-target RFS so that each sequential realization matches best the current target configuration. In that sense, a great variability in the target number between two successive time steps can be "accepted" by the PHD filter and the Poisson assumption might be less restrictive than it seems offhand.

Theorem 2.2 provides an insight on the shape of the posterior PHD. An important fact is that each measurement contributes "linearly" to the value of the posterior PHD. The ratio $\frac{\beta[\delta_z,\delta_x]}{\beta[\delta_z,1]}$ embodies the local contribution of measurement $z$ to the shape of the posterior PHD in point $x$, while $\frac{\int_{\mathcal{X}}\beta[\delta_z,\delta_x]dx}{\beta[\delta_z,1]}$ is the global contribution of measurement $z$ to the PHD, i.e. its contribution to the posterior estimated number of targets. It is easy to see the influence of the false alarm term on the contribution by considering the two following extreme cases:

1. If $z$ is "clearly" a false alarm, i.e. $\lambda^c_{k+1}c_{k+1}(z) \gg v_{\Xi_{k+1|k}}[p^d_{k+1}L^z_{k+1}]$, then by construction (proposition 2.3) $\beta[\delta_z,1] \simeq \lambda^c_{k+1}c_{k+1}(z)K_{\mathcal{Z}}$, thus the global contribution of measurement $z$ tends to be negligible $\left(\frac{\int_{\mathcal{X}}\beta[\delta_z,\delta_x]dx}{\beta[\delta_z,1]} \simeq \frac{\int_{\mathcal{X}}\beta[\delta_z,\delta_x]dx}{\lambda^c_{k+1}c_{k+1}(z)K_{\mathcal{Z}}} \ll 1\right)$.

2. Conversely, if $z$ is "clearly" a true measurement, i.e. $\lambda^c_{k+1}c_{k+1}(z) \ll v_{\Xi_{k+1|k}}[p^d_{k+1}L^z_{k+1}]$, then $\beta[\delta_z,1] \simeq v_{\Xi_{k+1|k}}[p^d_{k+1}L^z_{k+1}]K_{\mathcal{Z}} = \int_{\mathcal{X}}\beta[\delta_z,\delta_x]dx$, thus the global contribution of measurement $z$ tends to one $\left(\frac{\int_{\mathcal{X}}\beta[\delta_z,\delta_x]dx}{\beta[\delta_z,1]} \simeq 1\right)$.

More generally, the global contribution $\frac{\int_{\mathcal{X}}\beta[\delta_z,\delta_x]dx}{\beta[\delta_z,1]}$ is a real number between 0 and 1, increasing with the "credit" that can be granted to the measurement. If $z$ is likely to be a false alarm measurement, its global contribution is modest; the more the measurement can be "trusted", the higher its contribution is. In other words, the higher $\lambda^c_{k+1}c_{k+1}(z)$ is, the more measurement $z$ is implicitely considered as a false alarm.

The influence of the detection probability on the contribution is clear in the extreme case where a target lies outside the FOV, i.e. $p^d_{k+1}(x) = 0$. In this case, proposition 2.2 reduces to $v_{\Xi_{k+1|k+1}}(.|Z_{1:k+1}) = v_{\Xi_{k+1|k}}(.|Z_{1:k})$. This is expected: since a target in $x$ cannot be detected, no measurements can possibly stem from this target and the data update step does not provide new information in $x$; thus, the posterior PHD equals the predicted PHD. In any other case, though, the influence of the probability detection $p^d_{k+1}$ on the posterior PHD is less obvious. The general form of the local contribution $\frac{p^d_{k+1}(x)L^z_{k+1}(x)}{\lambda^c_{k+1}c_{k+1}(z)+v_{\Xi_{k+1|k}}[p^d_{k+1}L^z_{k+1}]}v_{\Xi_{k+1|k}}(.|Z_{1:k})$ suggests that a measurement in $z$ will sharpen the PHD around the highest probable origin $x_o = \arg\max_x p^d_{k+1}(x)L^z_{k+1}(x)$, provided that the likelihood function is discriminating enough (which is the case in practical situations, see chapter 4). It may be safely assumed that the detection probability function is much less discriminating than the likelihood function for usual sensors: that is, the local variation of the detection probability, aside from the edges of the FOV, is likely to be much slower than the variation of the likelihood function. In that sense, the influence of $p^d_{k+1}(.)$ on the local contribution is likely to be modest.

The influence of the detection probability on the global contribution should be studied in parallel with the contribution of the predicted PHD - that is, the past information on the target configuration. Consider a current measurement $z$, appearing in an region $R$ of the state space. If $z$ is inconsistent with past information on the targets ($v_{\Xi_{k+1|k}}$ is close to zero in $R$) and/or with the current FOV configuration ($p^d_{k+1}$ is close to zero in $R$), then $v_{\Xi_{k+1|k}}[p^d_{k+1}L^z_{k+1}] \simeq 0$ and thus the global contribution of $z$ is likely to be negligible. Conversely, if $z$ is consistent with known information regarding the targets and the sensors (typically, $z$ is close to a "large amount of PHD"), then $v_{\Xi_{k+1|k}}[p^d_{k+1}L^z_{k+1}]$ is likely do be dominant compared to the false alarm term, and the global contribution of $z$ tends to one $\left(\frac{\int_{\mathcal{X}} \beta[\delta_z,\delta_x]dx}{\beta[\delta_z,1]} \simeq 1\right)$. Consequently, measurements appearing "out of nowhere" are likely to be discarded before measurements appearing in the vicinity of previously detected targets. This "self-gating property" [Mahl 07b] is somewhat reassuring, although it has undesirable consequences for the exploration of unkown region of the state space.

## 2.3 Multi-sensor PHD filter

This section deals with the extension of the PHD filter data update equation to the multi-sensor case (theorem 2.2). Note that the time update equation does not involve any new measurements and therefore remains unchanged in the multi-sensor case. Note that the sensor order is arbitrary and need not be the same at each time step, nor does the sensor number. For clarity's sake, however, the sensor number is from now on assumed constant and equal to $S$.

### 2.3.1 Data update equation

#### Construction

The challenge is to find a tractable form of the Bayes data update equation (1.56) in the multi-sensor case:

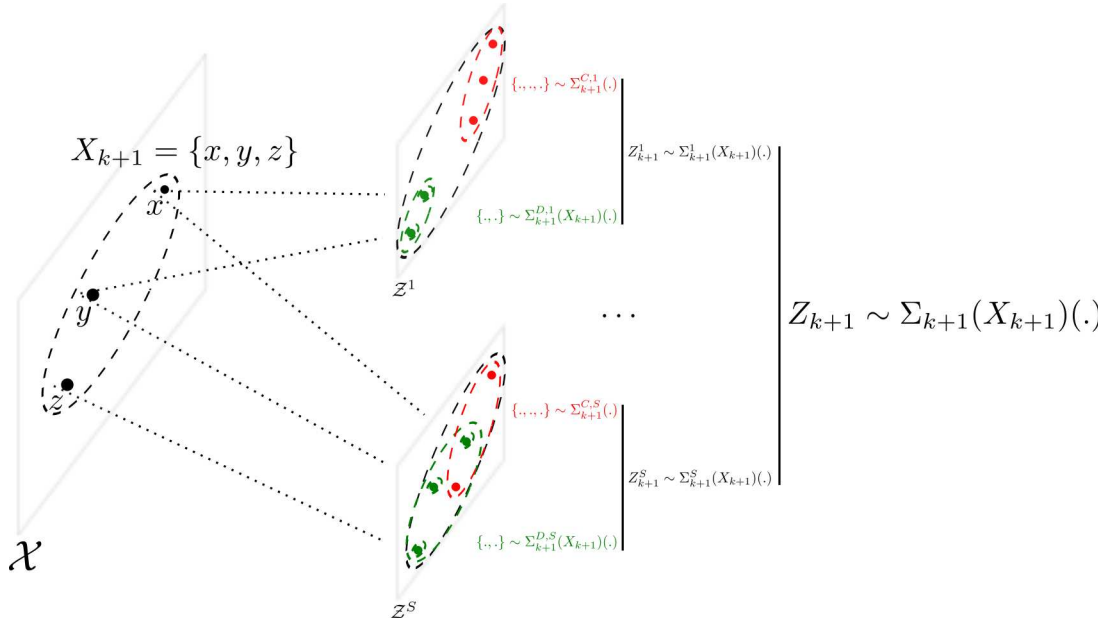
$$p_{\Xi_{k+1|k+1}}(.|Z_{1:k+1}) = \frac{p_{\Sigma_{k+1}}(Z_{k+1}|.)p_{\Xi_{k+1|k}}(.|Z_{1:k})}{\int_{\mathcal{F}(\mathcal{X})} p_{\Sigma_{k+1}}(Z_{k+1}|X)p_{\Xi_{k+1|k}}(X|Z_{1:k})\mu(dX)} \tag{2.30}$$

where:

- $\Xi_{k+1|k}$ is the (multi-target) predicted RFS at time $k+1$;
- $\Sigma_{k+1}$ is the (multi-measurement) observation RFS at time $k+1$;
- $\Xi_{k+1|k+1}$ is the (multi-target) posterior RFS at time $k+1$;

- $Z_{k+1} = \bigsqcup_{j=1}^{S} Z_{k+1}^{j}$ is the (multi-sensor) measurement set produced by all the sensors at time $k+1$.

**Figure 2.4** Example of multi-sensor observation RFS




The construction of the single-sensor observation RFS can be naturally extended to the multi-sensor case:

**Proposition 2.4.** *Assuming that, at time $k+1$:*

- *the observation process of each sensor is as described in proposition 2.2;*
- *the observation processes are statistically independent conditionally on the set $X_{k+1}$ of living targets.*

*then the observation RFS $\Sigma_{k+1}$ is the joint RFS:*

$$\Sigma_{k+1}(X) \stackrel{def}{=} \bigsqcup_{j=1}^{S} \Sigma_{k+1}^{j}(X) \tag{2.31}$$

*where $\Sigma_{k+1}^{j}(X)$ are the independent single-target observation RFSs. Besides, the probability distribution of the multi-sensor observation RFS exists and, for any multi-sensor measurement set $Z = \bigsqcup_{j=1}^{S} Z^{j}$:*

$$p_{\Sigma_{k+1}(X)}(Z) = \prod_{j=1}^{S} p_{\Sigma_{k+1}^{j}(X)}(Z^{j}) \tag{2.32}$$

The proof is straightforward by combining the equivalent result in the single-sensor case (proposition 2.2) and the property of joint RFSs (1.8).

Figure 2.4 shows an example of observation RFS based on a given set $X_{k+1}$. Note that target $z$ is missed by sensor 1 but detected by sensor $S$. Besides, three false alarms are produced by sensor 1 and two by sensor $S$.

The next step is the extension of the cross-term defintion to the multi-sensor case:

**Definition 2.6.** *Under the same assumptions as proposition 2.4, the multi-sensor cross-term $\beta_{k+1}$ is the functional defined, for any real-valued functions $h$ (resp. $g^j$, $j \in [1\ S]$) defined on $\mathcal{X}$ (resp. $\mathcal{Z}^j$, $j \in [1\ S]$) in $[0\ 1]$, by:*

$$\begin{aligned}\beta_{k+1}[g^1, ..., g^S, h] \overset{def}{=} & \sum_{j=1}^{S} (\lambda_{k+1}^{c,j} c_{k+1}^{j}[g^j] - \lambda_{k+1}^{c,j}) \\ & + v_{\Xi_{k+1|k}} \left[ h \left( \prod_{j=1}^{S} (1 - p_{k+1}^{d,j} + p_{k+1}^{d,j} f_{k+1}^{o,j}[g^j|.]) \right) \right] - v_{\Xi_{k+1|k}}[1] \end{aligned} \quad (2.33)$$

*where $v_{\Xi_{k+1|k}}[h]$ is the functional $v_{\Xi_{k+1|k}}[h] \overset{def}{=} \int_{\mathcal{X}} h(x) v_{\Xi_{k+1|k}}(x|Z_{1:k})dx$.*

For simplicity's sake, the time subscripts on cross-terms will be omitted when there is no ambiguity. The cross-term is a joint functional whose functional derivative (see definition 1.10) can be computed in functions defined on $\mathcal{X}$ or any $\mathcal{Z}^j$. In addition to the still valid notations (2.20) and (2.21) provided for the single-sensor cross-term, the following notations will be used from now on:

**Notation 2.2.** *For any $x \in \mathcal{X}$, any family of measurements $\{z^j\}_{j=1}^S$, $z^j \in \mathcal{Z}^j$, any subset $Z \subseteq \{z^j\}_{j=1}^S$:*

$$\beta[\delta_Z, \bar{g}, .] \overset{not}{=} \frac{\delta}{\delta Z} \beta[g^1, ..., g^S, .] \quad (2.34)$$

$$\beta[\delta_Z, .] \overset{not}{=} [\beta[\delta_Z, \bar{g}, .]]_{g^{1...S}=0} \quad (2.35)$$

*And, by convention:*

$$\beta[\delta_\emptyset, \bar{g}, .] \overset{not}{=} \beta[g^1, ..., g^S, .] \quad (2.36)$$

$$\beta[\delta_\emptyset, .] \overset{not}{=} \beta[0, ..., 0, .] \quad (2.37)$$

**Example 2.1.** *Suppose that there are* $S = 3$ *sensors. For* $j \in [1\ 3]$, *let* $z^j \in \mathcal{Z}^j$ *be any measurement from sensor* $s^j$, *and let* $x \in \mathcal{X}$ *be any target state. Then, for example:*

$$\begin{aligned}\beta[\delta_{\{z^1,z^3\}}, \bar{g}, h] &= \beta[\delta_{z^1}, g^2, \delta_{z^3}, h] \\ \beta[\delta_{\{z^2\}}, \delta_x] &= \beta[0, \delta_{z^2}, 0, \delta_x] \\ \beta[\delta_\emptyset, \bar{g}, 1] &= \beta[g^1, g^2, g^3, 1]\end{aligned}$$

**Proposition 2.5.** *For any* $x \in \mathcal{X}$, *any family of measurements* $\{z^j\}_{j=1}^S$, $z^j \in \mathcal{Z}^j$, *any subset* $Z \subseteq \{z^j\}_{j=1}^S$:

$$\beta[\delta_\emptyset, \bar{g}, \delta_x] = \prod_{j=1}^{S}(1 - p_{k+1}^{d,j}(x) + p_{k+1}^{d,j}(x) f_{k+1}^{o,j}[g^j|x]) v_{\Xi_{k+1|k}}(x|Z_{1:k}) K_{\mathcal{X}} \tag{2.38}$$

$$\beta[\delta_Z, \bar{g}, h] = \begin{cases} \lambda_{k+1}^{c,j_0} c_{k+1}^{j_0}(z^{j_0}) K_{\mathcal{Z}^{j_0}} \\ \quad + v_{\Xi_{k+1|k}}[h p_{k+1}^{d,j_0} L_{k+1}^{z^{j_0},j_0} K_{\mathcal{Z}^{j_0}} \prod_{z^j \neq z^{j_0}} (1 - p_{k+1}^{d,j} + p_{k+1}^{d,j} f_{k+1}^{o,j}[g^j|.])] \\ \qquad\qquad (Z = \{z^{j_0}\}) \\ v_{\Xi_{k+1|k}}[h \prod_{z^j \in Z} (p_{k+1}^{d,j} L_{k+1}^{z^j,j} K_{\mathcal{Z}^j}) \prod_{z^j \notin Z} (1 - p_{k+1}^{d,j} + p_{k+1}^{d,j} f_{k+1}^{o,j}[g^j|.])] \\ \qquad\qquad (|Z| \geqslant 2) \end{cases} \tag{2.39}$$

$$\begin{aligned}\beta[\delta_Z, \bar{g}, \delta_x] = &\prod_{z^j \in Z} (p_{k+1}^{d,j}(x) L_{k+1}^{z^j,j}(x) K_{\mathcal{Z}^j}) \\ &\times \prod_{z^j \notin Z} (1 - p_{k+1}^{d,j}(x) + p_{k+1}^{d,j}(x) f_{k+1}^{o,j}[g^j|x]) v_{\Xi_{k+1|k}}(x|Z_{1:k}) K_{\mathcal{X}}\end{aligned} \tag{2.40}$$

*Besides, setting* $g^1 = 0, ..., g^S = 0, h = 1$ *gives:*

$$\beta[\delta_\emptyset, \delta_x] = \prod_{j=1}^{S}(1 - p_{k+1}^{d,j}(x)) v_{\Xi_{k+1|k}}(x|Z_{1:k}) K_{\mathcal{X}} \tag{2.41}$$

$$\beta[\delta_Z, 1] = \begin{cases} \lambda_{k+1}^{c,j_0} c_{k+1}^{j_0}(z^{j_0}) K_{\mathcal{Z}^{j_0}} + v_{\Xi_{k+1|k}}[p_{k+1}^{d,j_0} L_{k+1}^{z^{j_0},j_0} K_{\mathcal{Z}^{j_0}} \prod_{z^j \neq z^{j_0}} (1 - p_{k+1}^{d,j})] \\ \qquad\qquad (Z = \{z^{j_0}\}) \\ v_{\Xi_{k+1|k}}[\prod_{z^j \in Z} (p_{k+1}^{d,j} L_{k+1}^{z^j,j} K_{\mathcal{Z}^j}) \prod_{z^j \notin Z} (1 - p_{k+1}^{d,j})] \qquad (|Z| \geqslant 2) \end{cases} \tag{2.42}$$

$$\beta[\delta_Z, \delta_x] = \prod_{z^j \in Z} (p_{k+1}^{d,j}(x) L_{k+1}^{z^j,j}(x) K_{\mathcal{Z}^j}) \prod_{z^j \notin Z} (1 - p_{k+1}^{d,j}(x)) v_{\Xi_{k+1|k}}(x|Z_{1:k}) K_{\mathcal{X}} \tag{2.43}$$

The proof is given in appendix A. Similarly to the single-sensor cross case, the functions $g^j$ (resp. $h$) can be seen as a "fuzzy" membership function on measurement space $\mathcal{Z}^j$ (resp. target space $\mathcal{X}$). Multi-sensor derivated cross-terms can be seen as "likelihoods" as well, conditonally on the predicted PHD $v_{\Xi_{k+1|k}}$, that:

- $\beta[\delta_\emptyset, \delta_x]$: a target is in state $x$ *and* is undetected;
- $\beta[\delta_Z, 1]$: a collection of measurements, whose *single* origin is unknown, are produced in points given by $Z$;
- $\beta[\delta_Z, \delta_x]$: a target is in state $x$, is the origin of measurements in $Z$ and is undetected by the remaining sensors;

As in the single-sensor case, pay attention to the fact that the term "likelihood" is an abuse of notation here.

Note that the false alarm terms vanish in $\beta[\delta_Z, h]$ if $Z$ contains at least two measurements. This is consistent with the observation model (proposition 2.4): the single-sensor observation processes being independent *conditionally on the states of the true targets*, there is no statistical link between the occurence of a false alarm in $z^i$ by sensor $i$ and the occurence of a false alarm in $z^j$ by sensor $j$. For example:

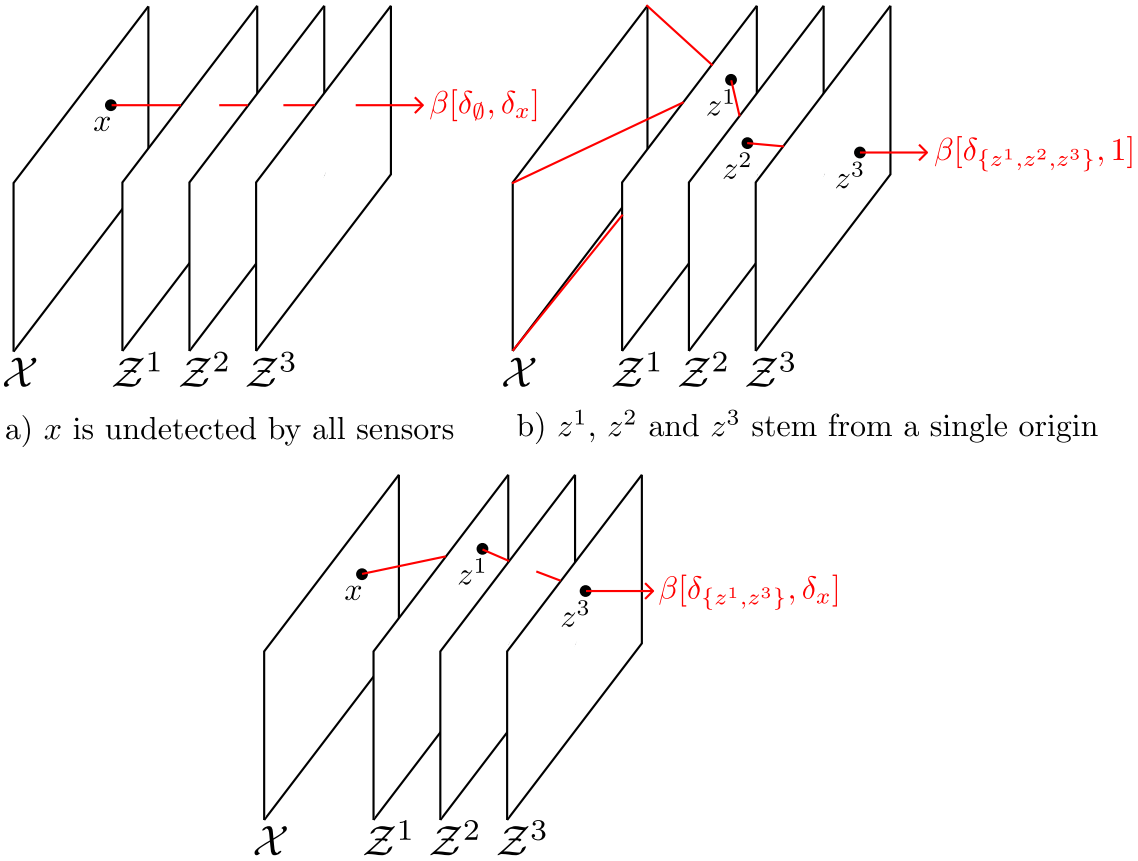
1. $\beta[\delta_{\{z^i\}}, 1]$ can be seen as the "likelihood" that a source, *whether* a target in an unknown state *or* a false alarm produced by sensor $i$, is the origin of measurement $z^i$.

2. $\beta[\delta_{\{z^i, z^j\}}, 1]$ can be seen as the "likelihood" that a *single* source, *necessarily* a target in an unknown state, is the origin of *both* measurements $z^i$ and $z^j$.

3. $\beta[\delta_{\{z^i\}}, 1]\beta[\delta_{\{z^j\}}, 1]$ can be seen as the "likelihood" that a source, *whether* a target in an unknown state *or* a false alarm produced by sensor $i$, is the origin of measurement $z^i$, while *another* source, *whether* a target in an unknown state *or* a false alarm produced by sensor $j$, is the origin of measurement $z^j$.

As in the single-sensor case, a cross-term derivated in several measurements from the *same* sensor (e.g. $\frac{\delta^2}{\delta z_2^i \delta z_1^i}\beta[g^1, ..., g^S, h]$) and/or in *several* targets (e.g. $\frac{\delta^2}{\delta x_2 \delta x_1}\beta[g^1, ..., g^S, h]$) vanishes. This is also consistent with the observation model (prop. 2.4) since a *single* true measurement cannot stem from *several* targets, nor can several true measurements from the *same* sensor stem from a *single* target.

**Figure 2.5** Illustration of the multi-sensor cross-term




a) $x$ is undetected by all sensors

b) $z^1$, $z^2$ and $z^3$ stem from a single origin

c) $z^1$ and $z^3$ stem from $x$

Once the observation RFS and the cross-term are properly extended, the PHD equivalent of the date update equation (2.16) in the multi-sensor case can be built as follows:

**Theorem 2.3.** *Under the assumptions given in proposition 2.4 and the additional assumption that the predicted RFS $\Xi_{k+1|k}$ is Poisson, the PHD filter data update equation is given by:*

$$v_{\Xi_{k+1|k+1}}(x|Z_{1:k+1}) = \frac{\left[\frac{\delta}{\delta x}\left(\frac{\delta}{\delta Z_{k+1}} e^{\beta[\delta_\emptyset, \bar{g}, h]}\right)\right]_{g^{1\ldots S}=0, h=1}}{\left[\frac{\delta}{\delta Z_{k+1}} e^{\beta[\delta_\emptyset, \bar{g}, h]}\right]_{g^{1\ldots S}=0, h=1}} K_{\mathcal{X}}^{-1} \tag{2.44}$$

$$= \frac{\left[\frac{\delta}{\delta x}\left(\frac{\delta}{\delta Z^1_{k+1}}\left(\ldots\left(\frac{\delta}{\delta Z^S_{k+1}} e^{\beta[g^1,\ldots,g^S,h]}\right)\ldots\right)\right)\right]_{g^{1\ldots S}=0, h=1}}{\left[\frac{\delta}{\delta Z^1_{k+1}}\left(\ldots\left(\frac{\delta}{\delta Z^S_{k+1}} e^{\beta[g^1,\ldots,g^S,h]}\right)\ldots\right)\right]_{g^{1\ldots S}=0, h=1}} K_{\mathcal{X}}^{-1} \tag{2.45}$$

*where:*

- $Z_{k+1}^j$ *is the single-sensor set of current measurements produced by sensor* $j$*;*

- $Z_{k+1} = \bigsqcup_{j=1}^{S} Z_{k+1}^j$ *is the multi-sensor set of current measurements.*

The proof is given in appendix A. Unfortunately, unlike the single-sensor case, there is no elegant analytical expression equivalent to the derivative forms (2.44) or (2.45). The exponential terms in (2.44) can be expanded by resolving the functional derivatives, but the resulting formula is of increasing complexity with the sensor number and/or the number of current measurements per sensor.

**Example 2.2.** *Suppose that there are* $S = 3$ *sensors. At current time - time subscripts are omitted for simplicity's sake - sensor* 1 *produces one measurement* $z_1^1$*, sensor* 2 *does not produce any measurement, sensor* 3 *produces two measurements* $z_1^3$ *and* $z_2^3$*. Applying theorem 2.3 gives:*

$$v_{\Xi_{k+1|k+1}}(x|Z_{1:k+1}) = \frac{\left[\frac{\delta}{\delta x}\left(\frac{\delta}{\delta z_1^1}\left(\frac{\delta}{\delta\{z_1^3,z_2^3\}}e^{\beta[g^1,g^2,g^3,h]}\right)\right)\right]_{g^j=0,h=1}}{\left[\frac{\delta}{\delta z_1^1}\left(\frac{\delta}{\delta\{z_1^3,z_2^3\}}e^{\beta[g^1,g^2,g^3,h]}\right)\right]_{g^j=0,h=1}} K_{\mathcal{X}}^{-1}$$

*A closer look at the denominator gives:*

$$\begin{aligned}
&\frac{\delta}{\delta z_1^1}\left(\frac{\delta}{\delta\{z_1^3,z_2^3\}}e^{\beta[g^1,g^2,g^3,h]}\right) = \frac{\delta}{\delta z_1^1}\left(\frac{\delta}{\delta z_1^3}\left(e^{\beta[g^1,g^2,g^3,h]}\beta[g^1,g^2,\delta_{z_2^3},h]\right)\right)\\
&= \frac{\delta}{\delta z_1^1}\left(e^{\beta[g^1,g^2,g^3,h]}(\beta[g^1,g^2,\delta_{z_1^3},h]\beta[g^1,g^2,\delta_{z_2^3},h])\right)\\
&= e^{\beta[g^1,g^2,g^3,h]}\Big(\beta[\delta_{z_1^1},g^2,\delta_{z_1^3},h]\beta[g^1,g^2,\delta_{z_2^3},h] + \beta[g^1,g^2,\delta_{z_1^3},h]\beta[\delta_{z_1^1},g^2,\delta_{z_2^3},h]\\
&\qquad + \beta[\delta_{z_1^1},g^2,g^3,h]\beta[g^1,g^2,\delta_{z_1^3},h]\beta[g^1,g^2,\delta_{z_2^3},h]\Big)
\end{aligned}$$

*Thus:*

$$\begin{aligned}
&\left[\frac{\delta}{\delta z_1^1}\left(\frac{\delta}{\delta\{z_1^3,z_2^3\}}e^{\beta[g^1,g^2,g^3,h]}\right)\right]_{g^j=0,h=1}\\
&= e^{\beta[\delta_\emptyset,1]}\left(\beta[\delta_{\{z_1^1,z_1^3\}},1]\beta[\delta_{z_2^3},1] + \beta[\delta_{z_1^3},1]\beta[\delta_{\{z_1^1,z_2^3\}},1] + \beta[\delta_{z_1^1},1]\beta[\delta_{z_1^3},1]\beta[\delta_{z_2^3},1]\right)
\end{aligned}$$

*Likewise, the numerator equals:*

$$\begin{aligned}
&\left[\frac{\delta}{\delta x}\left(\frac{\delta}{\delta z_1^1}\left(\frac{\delta}{\delta\{z_1^3,z_2^3\}}e^{\beta[g^1,g^2,g^3,h]}\right)\right)\right]_{g^j=0,h=1}\\
&= e^{\beta[\delta_\emptyset,1]}\Big(\beta[\delta_{\{z_1^1,z_1^3\}},\delta_x]\beta[\delta_{z_2^3},1]+\beta[\delta_{\{z_1^1,z_1^3\}},1]\beta[\delta_{z_2^3},\delta_x]+\beta[\delta_{z_1^3},\delta_x]\beta[\delta_{\{z_1^1,z_2^3\}},1]\\
&\quad+\beta[\delta_{z_1^3},1]\beta[\delta_{\{z_1^1,z_2^3\}},\delta_x]+\beta[\delta_{z_1^1},\delta_x]\beta[\delta_{z_1^3},1]\beta[\delta_{z_2^3},1]\\
&\quad+\beta[\delta_{z_1^1},\delta_x]\beta[\delta_{z_1^3},\delta_x]\beta[\delta_{z_2^3},1]+\beta[\delta_{z_1^1},\delta_x]\beta[\delta_{z_1^3},1]\beta[\delta_{z_2^3},\delta_x]\Big)\\
&\quad+\beta[\delta_\emptyset,\delta_x]\left[\frac{\delta}{\delta z_1^1}\left(\frac{\delta}{\delta\{z_1^3,z_2^3\}}e^{\beta[g^1,g^2,g^3,h]}\right)\right]_{g^j=0,h=1}
\end{aligned}$$

*Thus:*

$$\begin{aligned}
v_{\Xi_{k+1|k+1}}(x|Z_{1:k+1}) &= \beta[\delta_\emptyset,\delta_x]K_{\mathcal{X}}^{-1}\\
&\quad+\frac{\beta[\delta_{\{z_1^1,z_1^3\}},\delta_x]\beta[\delta_{z_2^3},1]+...+\beta[\delta_{z_1^1},1]\beta[\delta_{z_1^3},1]\beta[\delta_{z_2^3},\delta_x]}{\beta[\delta_{\{z_1^1,z_1^3\}},1]\beta[\delta_{z_2^3},1]+\beta[\delta_{z_1^3},1]\beta[\delta_{\{z_1^1,z_2^3\}},1]+\beta[\delta_{z_1^1},1]\beta[\delta_{z_1^3},1]\beta[\delta_{z_2^3},1]}K_{\mathcal{X}}^{-1}
\end{aligned}\tag{2.46}$$

Example 2.2 clearly shows how tedious the computation of the data update can be when the number of sensors and/or measurements is large enough. It also provides a more intuitive interpretation of the data update mechanism. The first cross-term "$\beta[\delta_\emptyset,\delta_x]$" weighs the event that a target lies in $x$ but is currently undetected, while the ratio accounts for the fact that a target lies in $x$ and is detected, that is, the origin of *at least* one current measurement. The numerators explores all the possible associations between $x$ and the current measurements; for example, "$\beta[\delta_{\{z_1^1,z_1^3\}},\delta_x]\beta[\delta_{z_2^3},1]$" weighs the event that a target lies in $x$, whose detection by both sensors 1 and 3 produces the measurements $z_1^1$ and $z_1^3$, while the last measurement $z_2^3$ stems from another source (whether a target or a false alarm). The denominator is a normalizing term and weighs the joint occurence of these three measurements.

Moreover, equation (2.46) provides an insight of the expanded expression of equation (2.44) - or, equivalently, (2.45) - in the general case. Similarly to the JPDA technique, a state point $x \in \mathcal{X}$ is updated as a weighted average of every possible measurement-to-target association, each derivated cross-term denoting one such association. The next paragraph focuses on the construction of the expanded expression - or *combinational* form - of the *derivative* form (2.44), because it is easier to manipulate for practical purposes and for the comparison with usual approximations of the multi-sensor PHD (see section 2.4).

### Combinational form

First of all, one must characterize the combination of measurement sets $Z$ on which the cross-terms $\beta[\delta_Z, .]$ appearing in the combinational form are derivated:

**Definition 2.7.** *For any subset $J \subseteq [1\ S]$, any family $\{Z^j\}_{j\in J}$ of finite subsets $Z^j \subset \mathcal{Z}^j$:*

*1. The (multi-measurement) term set $\mathcal{M}(Z^{j\in J}) \subset \mathcal{P}\left(\bigsqcup_{j\in J} Z^j\right)$ is given by:*

$$\mathcal{M}(Z^{j\in J}) \stackrel{def}{=} \bigcup_{I\subseteq J} \chi\left(\prod_{i\in I} Z^i\right) \tag{2.47}$$

*That is, a (multi-measurement) term $M \in \mathcal{M}(Z^{j\in J})$ is a set containing at most one measurement from each $Z^j$.*

*2. The signature $\varphi_{Z^{j\in J}}(.)$ is the function given by:*

$$\begin{array}{rccc} \varphi_{Z^{j\in J}}: & \mathcal{P}\left(\mathcal{P}\left(\bigsqcup_{j\in J} Z^j\right)\right) & \to & \mathbb{N} \\ & P & \mapsto & \prod_{z\in\bigsqcup_{j\in J} Z^j}\left(\sum_{P_i\in P} 1_{P_i}(z)\right) \end{array} \tag{2.48}$$

*Besides, $\varphi_{Z^{j\in J}}(\emptyset) \stackrel{def}{=} 0$ by convention. That is, $\varphi_{Z^{j\in J}}(P) = 1$ if and only if each measurement in $\bigsqcup_{j\in J} Z^j$ appears once and only once among all the sets $P_i$ in $P$.*

*3. The combinational term set $\mathcal{C}(Z^{j\in J})$ is given by:*

$$\mathcal{C}(Z^{j\in J}) \stackrel{def}{=} \left\{C \subseteq \mathcal{M}(Z^{j\in J}) \mid \varphi_{Z^{j\in J}}(C) = 1\right\} \tag{2.49}$$

*That is, a combinational term $C \in \mathcal{C}(Z^{j\in J})$ is a set of terms $C_i$ such that each measurement in $\bigsqcup_{j\in J} Z^j$ appears once and only once among all the terms $C_i$.*

*Besides, if $Z = \bigsqcup_{j\in J} Z^j$, $\mathcal{M}(Z)$ (resp. $\varphi_Z(.)$, $\mathcal{C}(Z)$) will denote $\mathcal{M}(Z^1, .., Z^J)$ (resp. $\varphi_{Z^1,...,Z^J}(.)$, $\mathcal{C}(Z^1, ..., Z^J)$) without ambiguity.*

**Example 2.3.** *Continuing example 2.2, in which* $S = 3$, $Z^1 = \{z_1^1\}$, $Z^2 = \emptyset$ *and* $Z^3 = \{z_1^3, z_2^3\}$, *gives:*

$$Z^1 \sqcup Z^2 \sqcup Z^3 = \{z_1^1, z_1^3, z_2^3\}$$

$$\mathcal{P}(Z^1 \sqcup Z^2 \sqcup Z^3) = \left\{\emptyset, \{z_1^1\}, \{z_1^3\}, \{z_2^3\}, \{z_1^1, z_1^3\}, \{z_1^1, z_2^3\}, \{z_1^3, z_2^3\}, \{z_1^1, z_1^3, z_2^3\}\right\}$$

$$\mathcal{M}(Z^1, Z^2, Z^3) = \left\{\{z_1^1\}, \{z_1^3\}, \{z_2^3\}, \{z_1^1, z_1^3\}, \{z_1^1, z_2^3\}\right\}$$

$$\mathcal{C}(Z^1, Z^2, Z^3) = \left\{\left\{\{z_1^1\}, \{z_1^3\}, \{z_2^3\}\right\}, \left\{\{z_1^1, z_1^3\}, \{z_2^3\}\right\}, \left\{\{z_1^1, z_2^3\}, \{z_1^3\}\right\}\right\}$$

*Note that, in the ratio in* (2.46)*:*

- *each cross-term is derivated in a multi-measurement term of* $\mathcal{M}(Z^1, Z^2, Z^3)$*;*
- *the cross-terms of a given product are derivated in the sets of a given combinational term of* $\mathcal{C}(Z^1, Z^2, Z^3)$.

*that is:*

$$v_{\Xi_{k+1|k+1}}(x|Z_{1:k+1}) = \beta[\delta_\emptyset, \delta_x]K_{\mathcal{X}}^{-1} + \frac{\displaystyle\sum_{C\in\mathcal{C}(Z_{k+1})}\sum_{C_i\in C}\left(\beta[\delta_{C_i}, \delta_x]\prod_{C_j\neq C_i}\beta[\delta_{C_j}, 1]\right)}{\displaystyle\sum_{C\in\mathcal{C}(Z_{k+1})}\prod_{C_i\in C}\beta[\delta_{C_i}, 1]}K_{\mathcal{X}}^{-1} \tag{2.50}$$

The proof of result (2.50) in the general case requires the following lemma:

**Lemma 2.1.** *For any sensor index* $s < S$, *any family* $\{Z^j\}_{j=1}^{s+1}$ *of finite subsets* $Z^j \subset \mathcal{Z}^j$, *with* $Z^{s+1} = \{z_i^{s+1}\}_{i=1}^{m^{s+1}}$*:*

$$\mathcal{C}(Z^{1:s+1}) = \bigcup_{C\in\mathcal{C}(Z^{1:s})}\ \bigcup_{n=0}^{\min(|C|,m^{s+1})}\ \bigcup_{\substack{I\subseteq[1\ \ m^{s+1}]\\ J\subseteq[1\ \ |C|]\\ |I|=|J|=n}}\ \bigcup_{\sigma\in Bij(I,J)} U_{I,J}^{\sigma}(Z^{s+1}, C) \tag{2.51}$$

*where* $Bij(I, J)$ *is the set of bijective functions from* $I$ *in* $J$ *and* $U_{I,J}^{\sigma}(Z^{s+1}, C) \in \mathcal{C}(Z^{1:s+1})$ *is the combinational term given by:*

$$U_{I,J}^{\sigma}(Z^{s+1}, C) \stackrel{def}{=} \left(\bigcup_{j\notin J}\{C_j\}\right) \cup \left(\bigcup_{i\in I}\left\{\{z_i^{s+1}\} \cup C_{\sigma(i)}\right\}\right) \cup \left(\bigcup_{i\notin I}\left\{\{z_i^{s+1}\}\right\}\right) \tag{2.52}$$

The proof is given in appendix A. Note that this lemma is useful for practical purposes because it shows that the combinational terms can be built recursively.

**Example 2.4.** *Continuing example 2.3 with* $s = 1$*, since* $m^1 = |Z^1| = 1$*,* $\mathcal{C}(Z^1)$ *is reduced to a single combinational term:*

$$\mathcal{C}(Z^1) = \{C\} = \left\{\left\{\{z_1^1\}\right\}\right\}$$

*Now with* $s = 2$*, since* $m^2 = |Z^2| = 0$*,* $\min(|C|, m^2) = 0$ *and thus:*

$$\mathcal{C}(Z^1, Z^2) = U^{Id}_{\emptyset,\emptyset}(\emptyset, C) = \{C_1\} = \left\{\left\{\{z_1^1\}\right\}\right\}$$

*Now with* $s = 3$*, since* $m^3 = |Z^3| = 2$*,* $\min(|C|, m^3) = 1$ *and thus:*

$$\mathcal{C}(Z^1, Z^2, Z^3) = U^{Id}_{\emptyset,\emptyset}(Z^3, C) \ \cup \ U^{1\leftrightarrow 1}_{\{1\},\{1\}}(Z^3, C) \ \cup \ U^{2\leftrightarrow 1}_{\{2\},\{1\}}(Z^3, C)$$

*with:*

$$\begin{aligned}
U^{Id}_{\emptyset,\emptyset}(Z^3, C) &= \{C_1\} \cup \left\{\{z_1^3\}\right\} \cup \left\{\{z_2^3\}\right\} = \left\{\{z_1^1\}, \{z_1^3\}, \{z_2^3\}\right\} \\
U^{1\leftrightarrow 1}_{\{1\},\{1\}}(Z^3, C) &= \left\{\{C_1 \cup \{z_1^3\}\right\} \cup \left\{\{z_2^3\}\right\} = \left\{\{z_1^1, z_1^3\}, \{z_2^3\}\right\} \\
U^{2\leftrightarrow 1}_{\{2\},\{1\}}(Z^3, C) &= \left\{\{C_1 \cup \{z_2^3\}\right\} \cup \left\{\{z_1^3\}\right\} = \left\{\{z_1^1, z_2^3\}, \{z_1^3\}\right\}
\end{aligned}$$

**Theorem 2.4.** *Under the assumptions given in proposition 2.4 and the additional assumption that the predicted RFS* $\Xi_{k+1|k}$ *is Poisson, the PHD filter data update equation is given by:*

$$v_{\Xi_{k+1|k+1}}(x|Z_{1:k+1}) = \beta[\delta_\emptyset, \delta_x]K_{\mathcal{X}}^{-1} + \frac{\displaystyle\sum_{C\in\mathcal{C}(Z_{k+1})}\sum_{C_i\in C}\left(\beta[\delta_{C_i}, \delta_x]\prod_{C_j\neq C_j}\beta[\delta_{C_j}, 1]\right)}{\displaystyle\sum_{C\in\mathcal{C}(Z_{k+1})}\prod_{C_i\in C}\beta[\delta_{C_i}, 1]}K_{\mathcal{X}}^{-1} \tag{2.53}$$

*where:*

- $Z_{k+1} = \bigsqcup_{j=1}^{S} Z_{k+1}^j$ *is the multi-sensor set of current measurements;*
- $\mathcal{C}(Z_{k+1})$ *is the set of combinational terms given by* (2.49)*.*

The proof is given in appendix A.

### Qualitative analysis

As for the single-sensor case, equation (2.53) provides some insight on the contribution of the current measurements in $Z_{k+1}$ to the posterior PHD. The notable difference with the single-sensor is that the contribution cannot be decoupled by individual measurements nor by individual sensors, keeping track of the contributions of the different measurements is therefore much more difficult. The influence of the false alarm terms can still be studied through the expression of the cross-terms since, according to proposition 2.5:

$$\begin{cases} \displaystyle\int_{\mathcal{X}} \beta[\delta_{\{z^{j_0}\}}, \delta_x] K_{\mathcal{X}}^{-1} dx + \lambda_{k+1}^{c,j_0} c_{k+1}^{j_0}(z^{j_0}) K_{\mathcal{Z}^{j_0}} = \beta[\delta_{\{z^{j_0}\}}, 1] & (C_i = \{z^{j_0}\}) \\ \displaystyle\int_{\mathcal{X}} \beta[\delta_{C_i}, \delta_x] K_{\mathcal{X}}^{-1} dx = \beta[\delta_{C_i}, 1] & (|C_i| \geqslant 2) \end{cases} \tag{2.54}$$

With the results above, it is easy to see that the global contribution of a measurement $z^{j_0}$ tends to zero if it is a false alarm, exactly as in the single-sensor case. Indeed $\lambda_{k+1}^{c,j_0} c_{k+1}^{j_0}(z^{j_0}) \gg 1$ and thus, according to the results above:

$$\begin{cases} \displaystyle\int_{\mathcal{X}} \beta[\delta_{\{z^{j_0}\}}, \delta_x] K_{\mathcal{X}}^{-1} dx \ll \beta[\delta_{\{z^{j_0}\}}, 1] & \\ \beta[\delta_{C_i}, 1] \ll \beta[\delta_{\{z^{j_0}\}}, 1] & (\{z^{j_0}\} \in C_i, |C_i| \geqslant 2) \end{cases} \tag{2.55}$$

Thus, by dividing both numerator and denominator of the ratio in equation (2.53) by $\beta[\delta_{\{z^{j_0}\}}, 1]$ and integrating the result over $\mathcal{X}$, all the combinational terms tend to zero but those where measurement $z^{j_0}$ is isolated, i.e. the combinational terms $C \in \mathcal{C}(Z_{k+1})$ of the form $C = \{\{z^{j_0}\}\} \cup C'$ where $C' \in \mathcal{C}(Z_{k+1} \setminus z^{j_0})$. And thus:

$$\begin{aligned} &\int_{\mathcal{X}} \left( \frac{\displaystyle\sum_{C \in \mathcal{C}(Z_{k+1})} \sum_{C_i \in C} \left( \beta[\delta_{C_i}, \delta_x] \prod_{C_j \neq C_j} \beta[\delta_{C_j}, 1] \right)}{\displaystyle\sum_{C \in \mathcal{C}(Z_{k+1})} \prod_{C_i \in C} \beta[\delta_{C_i}, 1]} K_{\mathcal{X}}^{-1} \right) dx \\ &\simeq \int_{\mathcal{X}} \left( \frac{\displaystyle\sum_{C \in \mathcal{C}(Z_{k+1} \setminus z^{j_0})} \sum_{C_i \in C} \left( \beta[\delta_{C_i}, \delta_x] \prod_{C_j \neq C_j} \beta[\delta_{C_j}, 1] \right)}{\displaystyle\sum_{C \in \mathcal{C}(Z_{k+1} \setminus z^{j_0})} \prod_{C_i \in C} \beta[\delta_{C_i}, 1]} K_{\mathcal{X}}^{-1} \right) dx \end{aligned}$$

This result is consistent with the observation model. If sensor $j$ produces a clear false alarm measurement $z$, then the influence of $z$ on the posterior PHD must be

minimal, but this must not preclude measurements from other sensors, even if they stem from a close point in state space, to be taken into account. In other words, if $z^1$ and $z^2$ fall in a region of the state space where 1 is known to produce a lot of false alarms, then the global contribution of $z^1$ on the posterior PHD should be discarded without compromising the contribution of $z^2$.

Intuitively, one may expect the opposite in case of true measurements. If $z^1$ and $z^2$ fall in a region of the state space where sensors 1 and 2 do not produce false measurements and do no miss detections, then the *joint* contribution of $z^1$ and $z^2$ should be around one, because $z^1$ and $z^2$ are almost surely two measurements from the *same* target. That is, the contribution of both $z^1$ and $z^2$ should be $\frac{1}{2}$ rather than 1, otherwise the target number would be overestimated. Results (2.54) yield:

$$\begin{aligned}
&\int_{\mathcal{X}} \left( \frac{\sum_{C\in\mathcal{C}(Z_{k+1})} \sum_{C_i\in C} \left( \beta[\delta_{C_i},\delta_x] \prod_{C_j\neq C_j} \beta[\delta_{C_j},1] \right)}{\sum_{C\in\mathcal{C}(Z_{k+1})} \prod_{C_i\in C} \beta[\delta_{C_i},1]} K_{\mathcal{X}}^{-1} \right) dx \\
&= \frac{\sum_{C\in\mathcal{C}(Z_{k+1})} \sum_{C_i\in C} \left( \int_{\mathcal{X}} \beta[\delta_{C_i},\delta_x] K_{\mathcal{X}}^{-1} dx \prod_{C_j\neq C_j} \beta[\delta_{C_j},1] \right)}{\sum_{C\in\mathcal{C}(Z_{k+1})} \prod_{C_i\in C} \beta[\delta_{C_i},1]} \\
&\leqslant \frac{\sum_{C\in\mathcal{C}(Z_{k+1})} |C] \prod_{C_j\in C} \beta[\delta_{C_j},1]}{\sum_{C\in\mathcal{C}(Z_{k+1})} \prod_{C_i\in C} \beta[\delta_{C_i},1]} \\
&\leqslant \max_C |C] \\
&\leqslant |Z_{k+1}|
\end{aligned}$$

As in the single-sensor case, the global contribution of a measurement $z$ never exceeds one. Now, consider the extreme case where there are no false alarms - $\lambda_{k+1}^{c,j_0} = 0$ - and there are no missed detections - $p_{k+1}^{d,j} = 1$ inside the FOV $F_{k+1}^j$. Then the first inequality above is an equality since $\int_{\mathcal{X}} \beta[\delta_{C_i},\delta_x] K_{\mathcal{X}}^{-1} dx = \beta[\delta_{C_i},1]$. Moreover:

1. If the FOVs are pairwise disjoint, according to the expression of the cross-terms (proposition 2.5) and provided that the likelihood functions are strictly positive, $\beta[\delta_{C_i},\delta_x] = 0$ if $|C_i| > 2$. Thus, the only remaining combinational term in the

global contribution is $C^0 = \bigcup_{j \in S} \bigcup_{i \in Z_{k+1}^j} \left\{ \{ z_i^j \} \right\}$ and therefore:

$$\begin{aligned}
&\int_{\mathcal{X}} \left( \frac{\sum_{C \in \mathcal{C}(Z_{k+1})} \sum_{C_i \in C} \left( \beta[\delta_{C_i}, \delta_x] \prod_{C_j \neq C_j} \beta[\delta_{C_j}, 1] \right)}{\sum_{C \in \mathcal{C}(Z_{k+1})} \prod_{C_i \in C} \beta[\delta_{C_i}, 1]} K_{\mathcal{X}}^{-1} \right) dx \\
&= \frac{|C^0] \prod_{C_j \in C^0} \beta[\delta_{C_j}, 1]}{\prod_{C_i \in C^0} \beta[\delta_{C_i}, 1]} \\
&= |C^0] \\
&= |Z_{k+1}|
\end{aligned}$$

2. Conversely, if the FOVs are all equal, then every true target is detected by each sensor, i.e. $|Z_{k+1}^j] = N$ where $N$ is the number of true targets. Besides, according to the expression of the cross-terms (proposition 2.5), $\beta[\delta_{C_i}, \delta_x] = 0$ if $|C_i| < S$. Thus, the only remaining combinational terms in the global contribution are those with $N$ multi-measurement terms $C_i$ with $S$ measurements each - recall from definition 2.7 that each one of the $|Z_{k+1}| = NS$ measurements appears once and only once in each combinational term $C$. Thus:

$$\begin{aligned}
&\int_{\mathcal{X}} \left( \frac{\sum_{C \in \mathcal{C}(Z_{k+1})} \sum_{C_i \in C} \left( \beta[\delta_{C_i}, \delta_x] \prod_{C_j \neq C_j} \beta[\delta_{C_j}, 1] \right)}{\sum_{C \in \mathcal{C}(Z_{k+1})} \prod_{C_i \in C} \beta[\delta_{C_i}, 1]} K_{\mathcal{X}}^{-1} \right) dx \\
&= \frac{\sum_{\substack{C \in \mathcal{C}(Z_{k+1}) \\ |C| = N}} |C] \prod_{C_j \in C} \beta[\delta_{C_j}, 1]}{\sum_{\substack{C \in \mathcal{C}(Z_{k+1}) \\ |C| = N}} \prod_{C_i \in C} \beta[\delta_{C_i}, 1]} \\
&= N \\
&= \frac{|Z_{k+1}|}{S}
\end{aligned}$$

These results were expected and are consistent with the observation model. If the FOV are pairwise disjoints, the $|Z_{k+1}|$ true measurements necessarily stem from

$|Z_{k+1}|$ true different targets, and the global contribution of $Z_{k+1}$ to the posterior PHD is $|Z_{k+1}|$, describing accurately the number of true targets. Conversely, if the FOVs are identical, each true target is the origin of $S$ true measurements - one per sensor - and the global contribution of $Z_{k+1}$ to the posterior PHD is once more the accurate number of true targets, i.e. $\frac{|Z_{k+1}|}{S}$. In conclusion, the global contribution of a measurement $z$ to the posterior PHD can be summarized as follows:

- if $z$ tends to be a false alarm, its global contribution tends to 0 regardless of the FOV configuration;
- if $z$ tends to be a true measurement, its global contribution tends to an upper bound equal to $\frac{1}{S}$ if the FOVs are identical and increasing up to 1 with the seperation of the FOVs.

There is not much to add regarding the influence of the detection probability on the posterior PHD that was not already discussed in the single-sensor case. If $x$ is outside all the FOVs, theorem 2.4 reduces to $v_{\Xi_{k+1|k+1}}(.|Z_{1:k+1}) = v_{\Xi_{k+1|k}}(.|Z_{1:k})$, as expected. As in the single-sensor case, the posterior is likely to sharpen around the measurement if the likelihood functions are discriminating enough. Two close measurements from two different sensors will "mutualize" their *local* contribution to a certain extent - i.e. the joint "sharpening effect" of the two measurement is likely to be more acute than the individual "sharpening effect" of each measurement - but with a limited effect on the *global* contribution as explained before.

### 2.3.2 Simplification by joint partitioning

Clearly, the computational cost of the data update equation (2.53) stems from the generation of the combinational terms $\mathcal{C}(Z_{k+1})$. Indeed, once these terms are known, computing each derivated cross-term is simple enough since they are explicitly constructed with common functions such as detection probabilities or single-target/single-measurement likelihood functions (see proposition 2.3). Is is also clear from lemma 2.1 that the computational cost of $\mathcal{C}(Z_{k+1})$ increases dramatically with the sensor number $S$ and/or the number of current measurements. This section proposes a partitioning method in order to simplify the data update equation (2.53) *without approximation*, in fact a simple rewriting of the data update equation such that the number of required combinational terms is significantly reduced.

The partitioning method is based on the FOVs configuration and is efficient in practical situations where the overlapping among the different FOVs is limited, for example when cameras are widely spread in the surveillance space such that the overlapping of more than three FOVs in a single point of the state space is unlikely at any time. It is based on the fact that many derivated cross-terms in $\mathcal{C}(Z_{k+1})$ are bound to vanish based on the FOV configuration, and therefore should not be computed.

The FOVs are properly defined as follows:

**Definition 2.8.** *For any sensor $j$, $j \in [1\ S]$, its field of view at time $k$ is the subset $F_k^j \subseteq \mathcal{X}$ defined as:*

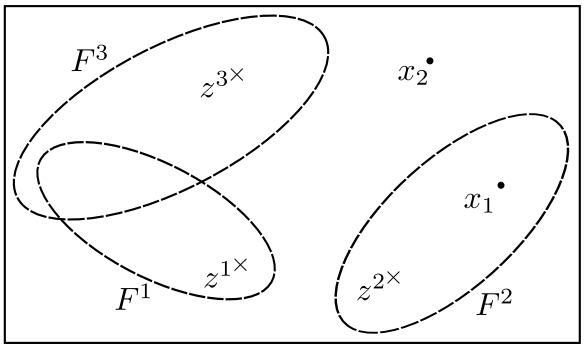
$$\forall x \in \mathcal{X}, x \in F_k^j \Leftrightarrow p_k^{d,j}(x) > 0 \tag{2.56}$$

From now, it is assumed that $F_k^j \neq \emptyset$ for any sensor $j$, at any time $k$. In the unlikely case that a sensor $j$ is to be "shut down" during a time step $k$ ($F_k^j = \emptyset$), this sensor is simply ignored for the time being and the remaining $S-1$ sensors are relabeled accordingly. Since the sensor number and the sensor order are arbitrary, there is no loss of generality.

Consider the following example:

**Example 2.5.** *Assuming that there are $S = 3$ sensors with current FOV configuration illustrated as follows (time subscripts are omitted for clarity's sake):*

**Figure 2.6** Simplification of some cross-terms based on the FOV configuration




*then some cross-terms are likely to simplify, for example:*

*Since $x_1 \notin F^1$, a target in state $x_1$ cannot be detected by sensor 1, and thus $\beta[\delta_{\{z^1\}}, \delta_{x_1}]$ should vanish.*

*Since $x_2 \notin F^2$, a target in state $x_2$ cannot be detected by sensor 2, and thus $\beta[\delta_{\{z^2\}}, \delta_{x_2}]$ should vanish. More generally, any cross-term derivated in $x_2$ but $\beta[\delta_{\emptyset}, \delta_{x_2}]$ should vanish.*

*Since $F^1 \cap F^2 = \emptyset$, no target may be detected by both sensors 1 and 2 and therefore no single source may be the origin of $z^1$ and $z^2$, thus $\beta[\delta_{\{z^1,z^2\}}, 1]$ should vanish.*

*Any single source at the origin of $z^1$ and $z^3$ must be detected by both sensors $s^1$ and $s^3$, thus $\beta[\delta_{\{z^1,z^3\}}, 1]$ should equal $\beta[\delta_{\{z^1,z^3\}}, 1_{F^1 \cap F^3}]$.*

This leads to the *joint partitioning* of sensor indices $[1\ S]$ and state space $\mathcal{X}$:

**Definition 2.9.** *Let the cross relation $\mathcal{R}_k$ be the reflexive, symmetric binary relation on sensors indices $[1\ S]$ defined by:*

$$\forall i, j \in [1\ S], i\mathcal{R}_k j \Leftrightarrow (F_k^i \cap F_k^j \neq \emptyset) \tag{2.57}$$

*and let $\mathcal{R}_k^+$ be its transitive closure.*

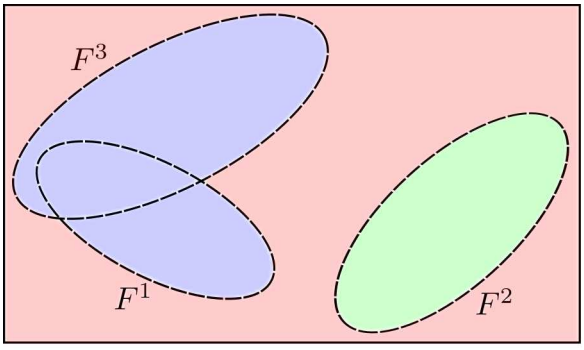
*Then, let $(S_k(p))_{p=1}^{P_k}$ be the sensor partition at time $k$, where $S_k(i)$ are the equivalence classes of $\mathcal{R}_k^+$, and $(T_k(p))_{p=0}^{P_k}$ be the target space partition at time $k$ defined by:*

$$T_k(p) \stackrel{def}{=} \begin{cases} \overline{\bigcup_{j=1}^{S} F_k^j} & (p = 0) \\ \bigcup_{j \in S_k(p)} F_k^j & (p \neq 0) \end{cases} \tag{2.58}$$

Note that the element order in $(S_k(p))_{p=1}^{P_k}$ is arbitrary but identical to the element order in $(T_k(p))_{p=0}^{P_k}$. For simplicity's sake, $(S_k(p))_{p=1}^{P_k}$ (resp. $(T_k(p))_{p=0}^{P_k}$) will be considered a partition of $[1\ S]$ (resp. $\mathcal{X}$), which is an abuse of notation since the elements are ordered and $T_k(0)$ may be empty if all points in the state space are covered by at least one FOV.

**Example 2.6.** *Continuing with example 2.5 leads to the following partitioning:*

**Figure 2.7** Illustration of the joint partitioning




$P = 2$ :

$T(0) = \mathcal{X} \setminus \bigcup_{j=1}^{3} F^j$

$S(1) = \{2\}$ $T(1) = F^2$

$S(2) = \{1, 3\}$ $T(2) = F^1 \cup F^3$

The *reduced cross-term functionals* $\beta_p$ are defined as follows:

**Definition 2.10.** *Let* $(S_{k+1}(p))_{p=1}^{P_{k+1}}$, $(T_{k+1}(p))_{p=0}^{P_{k+1}}$ *be the joint partitioning at time* $k+1$ *according to definition 2.9. Then, under the the same assumptions of proposition 2.4, the reduced cross-term* $\beta_{k+1,p}$, $1 \leqslant p \leqslant P_{k+1}$ *is the functional defined, for any real-valued functions* $h$ *(resp.* $g^j$, $j \in S_{k+1}(p)$*) defined on* $\mathcal{X}$ *(resp.* $\mathcal{Z}^j$, $j \in S_{k+1}(p)$*) in* $[0\ 1]$*, by:*

$$\begin{aligned}\beta_{k+1,p}[g^{j\in S_{k+1}(p)}, h] \overset{def}{=} & \sum_{j\in S_{k+1}(p)} (\lambda_{k+1}^{c,j} c_{k+1}^{j}[g^j] - \lambda_{k+1}^{c,j}) \\ & + v_{\Xi_{k+1|k}}\left[h\left(1_{T_{k+1}(p)} \prod_{j\in S_{k+1}(p)} (1 - p_{k+1}^{d,j} + p_{k+1}^{d,j} f_{k+1}^{o,j}[g^j|.])\right)\right] - v_{\Xi_{k+1|k}}[1_{T_{k+1}(p)}]\end{aligned} \tag{2.59}$$

*where* $v_{\Xi_{k+1|k}}[h]$ *is the functional* $v_{\Xi_{k+1|k}}[h] \overset{def}{=} \int_{\mathcal{X}} h(x) v_{\Xi_{k+1|k}}(x|Z_{1:k})dx$.

In other words, the reduced cross-term $\beta_{k+1,p}$ is the usual cross-term $\beta_{k+1}$ where only sensors in $S_{k+1}(p)$ and target states in $T_{k+1}(p)$ are considered. Clearly, all the expressions given in notation 2.2 and in proposition 2.5 are valid for $\beta_{k+1,p}$ once reduced to sensors in $S_{k+1}(p)$ and target states in $T_{k+1}(p)$. Likewise, definition 2.7 on combinational terms and the construction lemma 2.1 are valid when restricted to sensors in $S_{k+1}(p)$. As usual, time subscripts in the reduced cross-term will be omitted when there is no ambiguity.

The following proposition formalizes what was suggested in example 2.5:

**Proposition 2.6.** *Let* $(S_{k+1}(p))_{p=1}^{P_{k+1}}$, $(T_{k+1}(p))_{p=0}^{P_{k+1}}$ *be the joint partitioning at time* $k+1$*. For any* $x \in \mathcal{X}$*, any family of measurements* $\{z^j\}_{j=1}^{S}$, $z^j \in \mathcal{Z}^j$*, any subset* $J \subseteq [1\ S]$*:*

$$\beta[\delta_\emptyset, \bar{g}, \delta_x] = \begin{cases} \beta_p[\delta_\emptyset, \bar{g}, \delta_x] & (\exists p \in [1\ P_{k+1}], x \in T_{k+1}(p)) \\ v_{\Xi_{k+1|k}}(x) K_{\mathcal{X}} & (x \in T_{k+1}(0)) \end{cases} \tag{2.60}$$

$$\beta[\delta_{\{z^j, j\in J\}}, \bar{g}, h] = \begin{cases} \beta_p[\delta_{\{z^j, j\in J\}}, \bar{g}, h] & (\exists p \in [1\ P_{k+1}], J \subseteq S_{k+1}(p)) \\ 0 & (otherwise) \end{cases} \tag{2.61}$$

$$\beta[\delta_{\{z^j, j\in J\}}, \bar{g}, \delta_x] = \begin{cases} \beta_p[\delta_{\{z^j, j\in J\}}, \bar{g}, \delta_x] & (\exists p \in [1\ P_{k+1}], J \subseteq S_{k+1}(p), x \in T_{k+1}(p)) \\ 0 & (otherwise) \end{cases} \tag{2.62}$$

The proof is given in appendix A. The results of proposition 2.6 are quite intuitive and already illustrated in examples 2.5 and 2.6.

Equation (2.60): if a target $x$ belongs to the partition element $T_{k+1}(p)$, $p \neq 0$, then only (some) sensors in $T_{k+1}(p)$ may detect $x$. Thus, the "likelihood" that $x$ is undetected by all sensors - "$\beta[\delta_\emptyset, \delta_x]$" is the "likelihood" that $x$ is undetected by sensors from $T_{k+1}(p)$ - "$\beta_p[\delta_\emptyset, \delta_x]$". If $x$ is in $T_{k+1}(0)$ (the red area in figure 2.7), then the target is undetected with probability one and the cross-term reduces to the predicted PHD in $x$ - "$v_{\Xi_{k+1|k}}(x)K_{\mathcal{X}}$".

Equation (2.61): since the cross-term $\beta[\delta_{\{z^j, j\in J\}}, 1]$ weighs the association of measurements $z^j$, $j \in J$ to an unknown *single* source, it vanishes if sensors $j \in J$ do not *all* belong to the same partition element $S_{k+1}(p)$. If this is the case, then the single source either either lies in the combined FOV of these sensors, i.e. $T_{k+1}(p)$, or may eventually be a false alarm if there is only measurement, this joint event being weighted by $\beta_p[\delta_{\{z^j, j\in J\}}, 1]$ by construction.

Equation (2.62): since the cross-term $\beta[\delta_{\{z^j, j\in J\}}, \delta_x]$ weighs the association of measurements $z^j$, $j \in J$ to a *single* target, it vanishes if sensors $j \in J$ do not all belong to the same partition element $S_{k+1}(p)$ - i.e. no single source is "candidate" for this association, or if this the case but $x$ does not belong to the corresponding partition element $T_{k+1}(p)$ - i.e. there are "candidates" for this association, but $x$ is not. If $x$ does belong to $T_{k+1}(p)$, since all other "candidates" necessarily belong to $T_{k+1}(p)$ as well, $\beta[\delta_{\{z^j, j\in J\}}, \delta_x]$ reduces to $\beta_p[\delta_{\{z^j, j\in J\}}, \delta_x]$.

Considering the results of proposition 2.6, theorem 2.4 can then be simplified as follows:

**Theorem 2.5.** *Under the assumptions given in proposition 2.4 and the additional assumption that the predicted RFS $\Xi_{k+1|k}$ is Poisson, the PHD filter data update equation is given by:*

$$v_{\Xi_{k+1|k+1}}(x|Z_{1:k+1}) = \begin{cases} v_{\Xi_{k+1|k}}(x|Z_{1:k}) & (x \in T_{k+1}(0)) \\ \beta_p[\delta_\emptyset, \delta_x]K_{\mathcal{X}}^{-1} + \dfrac{\sum\limits_{C \in \mathcal{C}(Z_{k+1}^{(p)})} \sum\limits_{C_i \in C} \left( \beta_p[\delta_{C_i}, \delta_x] \prod\limits_{C_j \neq C_j} \beta_p[\delta_{C_j}, 1] \right)}{\sum\limits_{C \in \mathcal{C}(Z_{k+1}^{(p)})} \prod\limits_{C_i \in C} \beta_p[\delta_{C_i}, 1]} K_{\mathcal{X}}^{-1} & (x \in T_{k+1}(p), p \neq 0) \end{cases} \tag{2.63}$$

*where:*

- $(S_{k+1}(p))_{p=1}^{P_{k+1}}$, $(T_{k+1}(p))_{p=0}^{P_{k+1}}$ *is the current joint partitioning given by definition 2.9;*
- $Z_{k+1}^{(p)} = \bigsqcup_{j \in S_{k+1}(p)} Z_{k+1}^{j}$ *is the set of current measurements produced by sensors in $S_{k+1}(p)$;*
- $\mathcal{C}(Z_{k+1}^{(p)})$ *is the set of combinational terms given by* (2.49).

The proof is given in appendix A. From theorem 2.5 immediately follows the equivalent derivative form:

**Corollary 2.2.** *Under the assumptions given in proposition 2.4 and the additional assumption that the predicted RFS $\Xi_{k+1|k}$ is Poisson, the PHD filter data update equation is given by:*

$$v_{\Xi_{k+1|k+1}}(x|Z_{1:k+1}) = \begin{cases} v_{\Xi_{k+1|k}}(x|Z_{1:k}) & (x \in T_{k+1}(0)) \\ \dfrac{\left[ \frac{\delta}{\delta x} \left( \frac{\delta}{\delta Z_{k+1}^{(p)}} e^{\beta_p[\delta_\emptyset, \bar{g}, h]} \right) \right]_{g^{j \in S_{k+1}(p)}=0, h=1}}{\left[ \frac{\delta}{\delta Z_{k+1}^{(p)}} e^{\beta_p[\delta_\emptyset, \bar{g}, h]} \right]_{g^{j \in S_{k+1}(p)}=0, h=1}} K_{\mathcal{X}}^{-1} & (x \in T_{k+1}(p), p \neq 0) \end{cases} \tag{2.64}$$

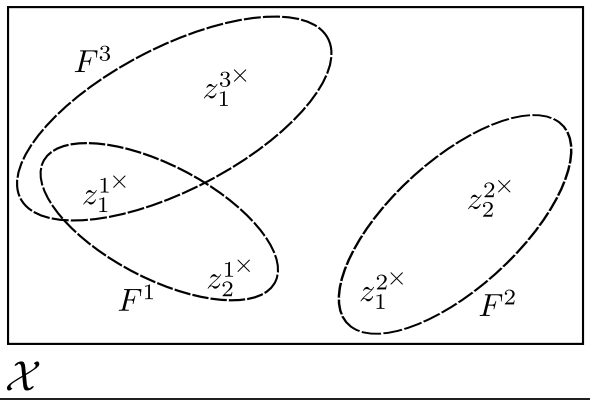
*where:*

- $(S_{k+1}(p))_{p=1}^{P_{k+1}}$, $(T_{k+1}(p))_{p=0}^{P_{k+1}}$ *is the current joint partitioning given by definition 2.9;*
- $Z_{k+1}^{(p)} = \bigsqcup_{j \in S_{k+1}(p)} Z_{k+1}^{j}$ *is the set of current measurements produced by sensors in* $S_{k+1}(p)$.

Even though the simplified data update equation (2.63) looks similar to the general case (2.53), it is considerably more practical when the FOV configuration is favorable enough for a partitioning. The following example shows that the gain can be significant, even in simple situations:

**Example 2.7.** *Assuming that the sensor number is* $S = 3$ *and that the current FOVs configuration and current measurements are as follows:*

**Figure 2.8** A FOV configuration favorable for partitioning




*then the sensor partition is* $(\{1,3\},\{2\})$ *and the combinational terms are:*

- $\mathcal{C}(Z^1, Z^3) = \Big\{\big\{\{z_1^1\},\{z_2^1\},\{z_1^3\}\big\},\big\{\{z_1^1, z_1^3\},\{z_2^1\}\big\},\big\{\{z_1^1\},\{z_2^1, z_1^3\}\big\}\Big\}$;
- $\mathcal{C}(Z^2) = \Big\{\big\{\{z_1^2\},\{z_2^2\}\big\}\Big\}$.

*That is, the simplified data update equation* (2.63) *requires the computation of* $|\mathcal{C}(Z^1, Z^3)| + |\mathcal{C}(Z^2)| = 4$ *combinational terms.*

*On the other hand, without partitioning, the combinational terms are* $\mathcal{C}(Z^1, Z^2, Z^3) = \Big\{\big\{\{z_1^1\},\{z_2^1\},\{z_1^2\},\{z_2^2\},\{z_1^3\}\big\},\big\{\{z_1^1, z_1^2, \},\{z_2^1\},\{z_2^2\},\{z_1^3\}\big\},...\Big\}$. *That is, the general data update equation* (2.53) *requires the computation of* $|\mathcal{C}(Z^1, Z^2, Z^3)| = 27$ *combinational terms.*

In the worst case, that is if the partitioning method described in definition 2.9 fails to split the sensors in more that one partition element, the computational cost of

(2.63) is actually *slightly worse* than (2.53) because it requires the computation of the partitioning itself in any case. However, as it will be shown later in chapter 4, the computational cost of the partitioning is light enough.

Regardless ot the computational gain given by the partitioning method, theorem 2.5 shows that the PHD can be updated *independently* on subparts of the state space - namely, the state partition elements $T_{k+1}(p)$. This is an important result, because it provides grounds for the design of hybrid PHD filters as trade-off between the costly but exact filter based on the combinational form (2.63) and usual approximations, such as the well-known iterated-corrector approximation (see section 2.4.2). Rather than using the iterated-corrector approximation on $\mathcal{X}$ with all the sensors, one can compute the joint partitioning and then decide, *independently on each partition element*, whether to use the exact data update equation or the iterated-corrector. It can be shown that the resulting hybrid filter performs at least as well as the iterated corrector; and the tweaking of the criteria allows a dynamical optimization of the filter's performance under the constraint of available computational power. This method, however, requires that one is able to estimate *a priori* the computational cost of the exact data update on a given partition element, presumably based on the element "size" (number of sensors and/or measurements). This will be discussed further in conclusion.

## 2.4 Common multi-sensor approximations

This section briefly describes several approximations of the multi-sensor PHD. The aim is not to compare of the filters on simulated data but rather to expose their strengths and weaknesses on a more theoretical level. Because the *iterated corrector* (see subsection 2.4.2 has already been implemented and seems to have good performances in detection and tracking problems [Mahl 10a], it will be compared on simulated data with the exact filter in chapter 4, while the other approximations are presented in this section for information purposes only and will not be studied further. An interesting study comparing the exact PHD filter - in the two-sensor case - and several approximation techniques on simulated data can be found in two recent papers (see [Naga 11b, Naga 11a] for more details).

### 2.4.1 Pseudo-sensor approximation

Arguably, the simplest way to face the multi-sensor issue is the *pseudo-sensor approximation*, in which all the sensors are encapsulated in a single "pseudo-sensor". That is, at every time step $k$, the single-sensor data update equation (2.28) is used

with the whole set of measurements $Z_{k+1} = \bigsqcup_{j=1}^{S} Z_{k+1}^{j}$ as input:

$$\tilde{v}_{\Xi_{k+1|k+1}}(.|Z_{1:k+1}) \\ \stackrel{def}{=} \left( 1 - \tilde{p}_{k+1}^{d}(x) + \sum_{j=1}^{S} \sum_{z^j \in Z_{k+1}^{j}} \frac{\tilde{p}_{k+1}^{d,j}(x) \tilde{L}_{k+1}^{z^j,j}(x)}{\tilde{\lambda}_{k+1}^{c,j} \tilde{c}_{k+1}^{j}(z) + v_{\Xi_{k+1|k}}[\tilde{p}_{k+1}^{d,j} \tilde{L}_{k+1}^{z,j}]} \right) v_{\Xi_{k+1|k}}(.|Z_{1:k}) \tag{2.65}$$

where $\tilde{p}_{k+1}^{d}$, $\tilde{p}_{k+1}^{d,j}$, $\tilde{\lambda}_{k+1}^{c,j} \tilde{c}_{k+1}^{j}$ $\tilde{L}_{k+1}^{.,j}$ are the pseudo-functions describing the mechanisms of the pseudo-sensor. Mahler [Mahl 03b, Mahl 03c] proposed such an approximation where the pseudo-functions are defined as follows:

- $\tilde{p}_{k+1}^{d}(.) \stackrel{def}{=} 1 - \prod_{j=1}^{S}(1 - p_{k+1}^{d,j}(.))$;
- $\tilde{p}_{k+1}^{d,j}(.) \stackrel{def}{=} p_{k+1}^{d,j}(.) \frac{\tilde{p}_{k+1}^{d}(.)}{\sum_{j=1}^{S} p_{k+1}^{d,j}(.)}$;
- $\tilde{\lambda}_{k+1}^{c,j} \tilde{c}_{k+1}^{j}(.) \stackrel{def}{=} \lambda_{k+1}^{c,j} c_{k+1}^{j}(.)$;
- $\tilde{L}_{k+1}^{.,j} \stackrel{def}{=} L_{k+1}^{.,j}(.)$.

Another pseudo-sensor approximation leads from the exact data update equation (2.44) with the additional assumption that a target $x$ is the source of at most one measurement among *all* the measurement sets $Z_{k+1}^{j}$, i.e. $\beta[\delta_Z, .] = 0$ if $|Z| > 1$. Indeed, with this new assumption:

$$\tilde{v}_{\Xi_{k+1|k+1}}(x|Z_{1:k+1}) = \frac{\left[\frac{\delta}{\delta x}\left(\frac{\delta}{\delta Z_{k+1}} e^{\beta[\delta_\emptyset, \bar{g}, h]}\right)\right]_{g^{1...S}=0, h=1}}{\left[\frac{\delta}{\delta Z_{k+1}} e^{\beta[\delta_\emptyset, \bar{g}, h]}\right]_{g^{1...S}=0, h=1}} K_{\mathcal{X}}^{-1} \qquad (\beta[\delta_Z, .] = 0, |Z| \geqslant 1)$$

$$= \frac{\left[\frac{\delta}{\delta x}\left(e^{\beta[\delta_\emptyset, \bar{g}, h]} \prod_{j=1}^{S} \prod_{z \in Z_{k+1}^{j}} \left(\beta[\delta_{\{z\}}, \bar{g}, h]\right)\right)\right]_{g^{1...S}=0, h=1}}{\left[e^{\beta[\delta_\emptyset, \bar{g}, h]} \prod_{j=1}^{S} \prod_{z \in Z_{k+1}^{j}} \left(\beta[\delta_{\{z\}}, \bar{g}, h]\right)\right]_{g^{1...S}=0, , h=1}} K_{\mathcal{X}}^{-1}$$

$$= \frac{\beta[\delta_\emptyset, \delta_x] e^{\beta[\delta_\emptyset, 1]} \prod_{j=1}^{S} \prod_{z \in Z_{k+1}^{j}} \left(\beta[\delta_{\{z\}}, 1]\right)}{e^{\beta[\delta_\emptyset, 1]} \prod_{j=1}^{S} \prod_{z \in Z_{k+1}^{j}} \left(\beta[\delta_{\{z\}}, 1]\right)} K_{\mathcal{X}}^{-1} \\ + \frac{e^{\beta[\delta_\emptyset, 1]} \left(\prod_{j=1}^{S} \prod_{z \in Z_{k+1}^{j}} \left(\beta[\delta_{\{z\}}, 1]\right)\right) \sum_{j=1}^{S} \sum_{z \in Z_{k+1}^{j}} \left(\frac{\beta[\delta_{\{z\}}, \delta_x]}{\beta[\delta_{\{z\}}, 1]}\right)}{e^{\beta[\delta_\emptyset, 1]} \prod_{j=1}^{S} \prod_{z \in Z_{k+1}^{j}} \left(\beta[\delta_{\{z\}}, 1]\right)} K_{\mathcal{X}}^{-1}$$

$$= \beta[\delta_\emptyset, \delta_x] K_{\mathcal{X}}^{-1} + \sum_{j=1}^{S} \sum_{z \in Z_{k+1}^{j}} \frac{\beta[\delta_{\{z\}}, \delta_x]}{\beta[\delta_{\{z\}}, 1]} K_{\mathcal{X}}^{-1}$$

Which gives, using proposition 2.5:

$$\begin{aligned}
&\tilde{v}_{\Xi_{k+1|k+1}}(x|Z_{1:k+1}) \\
&= \left( \prod_{j=1}^{N} (1 - p_{k+1}^{d,j}(x)) \right. \\
&\quad \left. + \sum_{j=1}^{S} \sum_{z \in Z_{k+1}^{j}} \frac{\left( \prod_{i \neq j} (1 - p_{k+1}^{d,i}(x)) \right) p_{k+1}^{d,j}(x) L_{k+1}^{z,j}(x)}{\lambda_{k+1}^{c,j} c_{k+1}^{j}(z) + v_{\Xi_{k+1|k}} \left[ \left( \prod_{i \neq j} (1 - p_{k+1}^{d,i}) \right) p_{k+1}^{d,j} L_{k+1}^{z,j} \right]} \right) v_{\Xi_{k+1|k}}(x|Z_{1:k})
\end{aligned}$$

This leads to another pseudo-sensor approximation, quite similar to Mahler's:

- $\tilde{p}_{k+1}^{d}(.) \overset{def}{=} 1 - \prod_{j=1}^{S} (1 - p_{k+1}^{d,j}(.))$;
- $\tilde{p}_{k+1}^{d,j}(.) \overset{def}{=} p_{k+1}^{d,j}(.) \prod_{i \neq j} (1 - p_{k+1}^{d,i}(.))$;
- $\tilde{\lambda}_{k+1}^{c,j} \tilde{c}_{k+1}^{j}(.) \overset{def}{=} \lambda_{k+1}^{c,j} c_{k+1}^{j}(.)$;
- $\tilde{L}_{k+1}^{.,j} \overset{def}{=} L_{k+1}^{.,j}(.)$.

The obvious advantage of pseudo-sensor approximations are their simplicity. However, whether it is by construction in Mahler's approximation or explicity stated by the additional assumption in the approximation above, the independence of the single-sensor processes is violated since it implies that a target may not be the origin of more than one measurement. Thus, while the exact multi-sensor PHD correctly associates up to $S$ measurements per target - one per sensor - at any time step, the limited pseudo-sensor framework fails to do so and considers only single measurement to single target associations, any additional measurements being considered as false alarms [Mahl 03b, Mahl 03c]. Clearly, pseudo-sensor approximations are quite constraining and are likely to perform poorly in areas where FOVs are overlapping.

### 2.4.2 Sequential approximation

The aim of the sequential approximation is to bypass the multi-sensor issue by dealing with each sensor *separately* at each time step; that is, using a sequence of $S$ single-sensor data update steps (2.27), namely once per sensor, rather than using the multi-sensor data update step (2.44) once. More precisely, the *iterated-corrector approximation* (ICA) is given by (adapted from [Mahl 03a, Mahl 10a]):

**Definition 2.11.** *For any permutation $\sigma : [1\ S] \to [1\ S]$, any current measurement sets $(Z_{k+1}^j)_{j=1}^S$, $Z^j \subset \mathcal{Z}^j$, any positive function $v(.)$ on $\mathcal{X}$, the corrector $C_{k+1}^{\sigma,j}(.,v)$, where $j \in [1\ S]$, is the function on $\mathcal{X}$ given by:*

$$C_{k+1}^{\sigma,j}(.,v) \stackrel{def}{=} 1 - p_{k+1}^{d,\sigma(j)}(.) + \sum_{z \in Z_{k+1}^{\sigma(j)}} \frac{p_{k+1}^{d,\sigma(j)}(.) L_{k+1}^{z,\sigma(j)}(.)}{\lambda_{k+1}^{c,\sigma(j)} c_{k+1}^{\sigma(j)}(z) + v[p_{k+1}^{d,\sigma(j)} L_{k+1}^{z,\sigma(j)}]} \tag{2.66}$$

*Then, the sequence of iterated approximations $(v_{k+1|k}^{\sigma,j})_{j=0}^S$ is the sequence of positive functions on $\mathcal{X}$ given by:*

$$v_{k+1|k}^{\sigma,0}(.) \stackrel{def}{=} v_{\Xi_{k+1|k}}(.|Z_k) \tag{2.67}$$

$$\forall j \in [1\ S],\ v_{k+1|k}^{\sigma,j}(.) \stackrel{def}{=} C_{k+1}^{\sigma,j}(., v_{k+1|k}^{\sigma,j-1}) \tag{2.68}$$

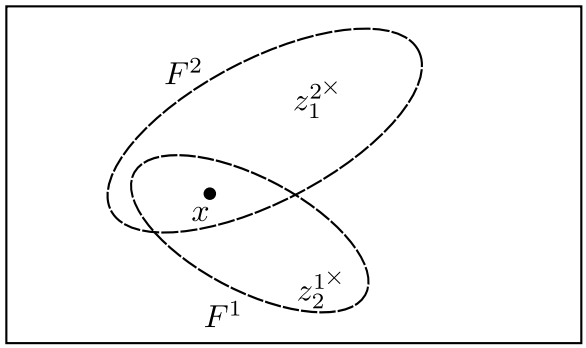
*The iterated approximation of the multi-sensor data update equation, respective to sensor order $\sigma(1), .., \sigma(S)$, is given by:*

$$\tilde{v}_{\Xi_{k+1|k+1}}^{\sigma}(.|Z_{1:k+1}) \stackrel{def}{=} v_{k+1|k}^{\sigma,S}(.) \tag{2.69}$$

In other words, the ICA proceeds with each sensor *in a given order*, applying the single-sensor data udpate equation (2.29) with the density from the previous iteration as input. Although slightly more complicated to implement than pseudo-sensor approximations, the ICA is more faithful to the multi-sensor model assumptions and, as it will be shown later on simulated data (see chapter 4), it seems to be fairly accurate (when compared to the exact multi-sensor data update) in detection and tracking problems with a limited number of sensors. However, the ICA suffers some flaws in its design [Mahl 09a]. First, the validity of the approximated posterior PHD $\tilde{v}_{\Xi_{k+1|k+1}}(.|Z_{1:k+1})$ in (2.69) is, by construction, based on the validity of the sequence of the $S$ single-sensor data update steps which, according to theorem 2.2, requires that every intermediate approximation $v_{k+1|k}^{\sigma,j}(.)$ can be seen as the PHD of a Poisson RFS, which can be difficult to ascertain in practical situations. The second issue, more obvious, is the asymmetry of the sensors in the ICA: in the general case, $\tilde{v}_{\Xi_{k+1|k+1}}^{\sigma_1}(.|Z_{1:k+1}) \neq \tilde{v}_{\Xi_{k+1|k+1}}^{\sigma_2}(.|Z_{1:k+1})$ if $\sigma_1 \neq \sigma_2$. This can be easily illustrated a simple example:

**Example 2.8.** *Assuuming that the sensor number is $S = 2$ and that the current FOV configuration and current measurements are as follows:*

**Figure 2.9** The ICA on a simple example




*Then, the exact posterior PHD given by theorem 2.4 is:*

$$\begin{aligned} &v_{\Xi_{k+1|k+1}}(x|Z_{1:k+1}) \\ &= \beta[\delta_\emptyset, \delta_x] + \frac{\beta[\delta_{\{z^1\}}, \delta_x]\beta[\delta_{\{z^2\}}, 1] + \beta[\delta_{\{z^1\}}, 1]\beta[\delta_{\{z^2\}}, \delta_x] + \beta[\delta_{\{z^1,z^2\}}, \delta_x]}{\beta[\delta_{\{z^1\}}, 1]\beta[\delta_{\{z^2\}}, 1] + \beta[\delta_{\{z^1,z^2\}}, 1]} \end{aligned} \tag{2.70}$$

*The approximation given by the ICA with sensor order* $1 \to 2$ *(*$\sigma = Id$*) is:*

$$\tilde{v}^{Id}_{\Xi_{k+1|k+1}}(x|Z_{1:k+1}) = \beta[\delta_\emptyset, \delta_x] + \frac{\beta^1[\delta_{\{z^1\}}, \delta_x]}{\beta^1[\delta_{\{z^1\}}, 1]} + \frac{\beta^1[\delta_{\{z^1\}}, 1]\beta[\delta_{\{z^2\}}, \delta_x] + \beta[\delta_{\{z^1,z^2\}}, \delta_x]}{\beta^1[\delta_{\{z^1\}}, 1]\beta[\delta_{\{z^2\}}, 1] + \beta[\delta_{\{z^1,z^2\}}, 1]} \tag{2.71}$$

*while the approximation given by the ICA with sensor order* $2 \to 1$ *(*$\sigma = \tau_{12}$*) is:*

$$\tilde{v}^{\tau_{12}}_{\Xi_{k+1|k+1}}(x|Z_{1:k+1}) = \beta[\delta_\emptyset, \delta_x] + \frac{\beta^2[\delta_{\{z^2\}}, \delta_x]}{\beta^2[\delta_{\{z^2\}}, 1]} + \frac{\beta[\delta_{\{z^1\}}, \delta_x]\beta^2[\delta_{\{z^2\}}, 1] + \beta[\delta_{\{z^1,z^2\}}, \delta_x]}{\beta[\delta_{\{z^1\}}, 1]\beta^2[\delta_{\{z^2\}}, 1] + \beta[\delta_{\{z^1,z^2\}}, 1]} \tag{2.72}$$

*where* $\beta^j[g^j, h] \stackrel{def}{=} \lambda^{c,j}_{k+1} c^j_{k+1}[g^j] - \lambda^{c,j}_{k+j} + v_{\Xi_{k+1|k}}\left[h(1 - p^{d,j}_{k+1} + p^{d,j}_{k+1} f^{o,j}_{k+1}[g^j|.])\right] - v_{\Xi_{k+1|k}}[1]$ *is the cross-term restricted to sensor* $s^j$*.*

Example 2.8 illustrates the approximation behind the ICA. With the sensor order $1 \to 2$, the corrrector updates the predicted PHD with measurements from sensor 1 - namely, $z^1$ - without considering the fact that another sensor 2 produced measurements *in the same time*. Thus, the first ratio in equation (2.71) $\frac{\beta^1[\delta_{\{z^1\}}, \delta_x]}{\beta^1[\delta_{\{z^1\}}, 1]}$ completely ignores the second sensor - recall that $\beta^1[\delta_{\{z^1\}}, \delta_x] = p^{d,1}_{k+1}(x) L^{z^1,1}_{k+1}(x) v_{\Xi_{k+1|k}}(x|Z_{1:k})$ while $\beta[\delta_{\{z^1\}}, \delta_x] = p^{d,1}_{k+1}(x) L^{z^1,1}_{k+1}(x)(1 - p^{d,2}_{k+1}(x)) v_{\Xi_{k+1|k}}(x|Z_{1:k})$. As shown in equation (2.71), the initial error propagates in the future ratios.

Clearly, the ICA's quality is likely to decrease with the number of sensors and/or measurements involved in the data update step, but quantifying the approximation with respect to the exact multi-sensor data seems quite challenging. Besides, there is no easy way to select, to the author's knowledge, the optimal sensor order for the ICA. We detailed in [Dela 10] a more explicit recursive expression of the ICA; it seems that the quality of the approximation increases with the sensor order, i.e. measurements from the last sensors are generally "better considered" than those from the first sensors. This suggests that a sound choice for the sensor order is to order them by increasing "productivity" (i.e. increasing number of current measurements), but no systematic rule could be derived. This will be illustrated on simulated data in chapter 4.

### 2.4.3 Product approximation

The *product approximation* (PA) aim at bypassing the asymmetry issue in the ICA by approximating the multi-sensor data update as a product of single-sensor correctors (adapted from [Mahl 09a]):

$$\tilde{v}^{K_{Z^1_{k+1},\dots,Z^S_{k+1}}}_{\Xi_{k+1|k+1}}(.|Z_{1:k+1}) \overset{def}{=} K_{Z^1_{k+1},\dots,Z^S_{k+1}} \prod_{j=1}^{S} (C^j_{k+1}(.|Z^j_{k+1}))v_{\Xi_{k+1|k}}(.|Z_{1:k}) \tag{2.73}$$

where:

- $C^j_{k+1}(.|Z^j_{k+1}) \overset{def}{=} 1 - p^{d,j}_{k+1}(x) + \sum_{z \in Z^j_{k+1}} \frac{p^{d,j}_{k+1}(x) L^{z,j}_{k+1}(x)}{\lambda^{c,j}_{k+1} c^j_{k+1}(z) + v_{\Xi_{k+1|k}}[p^{d,j}_{k+1} L^{z,j}_{k+1}]}$ is the corrector from sensor $j$;
- $K_{Z^1_{k+1},\dots,Z^S_{k+1}}$ is a constant, symmetric with respect to the sensors.

In other words, the principle of PAs is to encapsulate all the coupling effects between sensors in a single term. Clearly, the quality of a PA is based on the proper choice of the coupling term. In [Mahl 09a] Mahler explains that this method is not well adapted to the PHD yet. In the construction of the coupling term he had to make an assumption on the densities $p_{\Xi_{k+1|k}}(.|Z^j_{k+1})$ and came to the conclusion that:

- if they are assumed to be PHDs from cluster RFSs, the resulting coupling constant is intractable;
- if they are assumed to be PHDs from Poisson RFSs, the resulting coupling constant is 1, but $\tilde{v}^1_{\Xi_{k+1|k+1}}(.|Z_{1:k+1})$ is a poor approximation.

The PA with the coupling term equal to one (that we mentioned as *product approximation* in [Dela 10]) can be compared with the ICA in the following example:

**Example 2.9.** *With the same configuration and notations as example 2.8, the product approximation with* $K_{Z^1_{k+1},\dots,Z^S_{k+1}} = 1$ *is given by:*

$$\begin{aligned}&\tilde{v}^1_{\Xi_{k+1|k+1}}(x|Z_{1:k+1})\\&= \beta[\delta_\emptyset,\delta_x] + \frac{\beta[\delta_{\{z^1\}},\delta_x]\beta^2[\delta_{\{z^2\}},1] + \beta^1[\delta_{\{z^1\}},1]\beta[\delta_{\{z^2\}},\delta_x] + \beta[\delta_{\{z^1,z^2\}},\delta_x]}{\beta^1[\delta_{\{z^1\}},1]\beta^2[\delta_{\{z^2\}},1]}\end{aligned} \tag{2.74}$$

Example 2.9 shows the PA with $K = 1$ fails at considering cross-sensor measurement associations (the cross-term $\beta[\delta_{\{z^1,z^2\}},\delta_x]$ is actually a calculus "side-effect", what is important here is that the cross-term $\beta[\delta_{\{z^1,z^2\}},1]$ does not appear in the denominator). Mahler remarked [Mahl 09a] that the PA with $K = 1$ does not reduce to the multi-sensor/single-target Bayes filter in the trivial case where there are no false alarm, no missed detections and a single true target, while the ICA does. Indeed:

**Example 2.10.** *Continuing examples 2.8 and 2.9 with the additional assumptions that* $\lambda^{c,1}_{k+1} = \lambda^{c,2}_{k+1} = 0$, $p^{d,1}_{k+1}(.) = p^{d,2}_{k+1}(.) = 1$*:*

- *the exact posterior is* $v_{\Xi_{k+1|k+1}}(x|Z_{1:k+1}) = \frac{\beta[\delta_{\{z^1,z^2\}},\delta_x]}{\beta[\delta_{\{z^1,z^2\}},1]}$;
- *the ICAs give* $\tilde{v}^{Id}_{\Xi_{k+1|k+1}}(x|Z_{1:k+1}) = \tilde{v}^{T12}_{\Xi_{k+1|k+1}}(x|Z_{1:k+1}) = \frac{\beta[\delta_{\{z^1,z^2\}},\delta_x]}{\beta[\delta_{\{z^1,z^2\}},1]}$;
- *the PA with* $K = 1$ *gives* $\tilde{v}^1_{\Xi_{k+1|k+1}}(x|Z_{1:k+1}) = \frac{\beta[\delta_{\{z^1,z^2\}},\delta_x]}{\beta_1[\delta_{\{z^1\}},1]\beta_2[\delta_{\{z^2\}},1]}$.

It is not clear how worse this PA is compared to the ICA but, based on a more explicit expression of the PA we built in [Dela 10], it seems that the ICA is generally better, although comparisons on simulated data should be quite useful to answer this question.

## 2.5 Conclusion

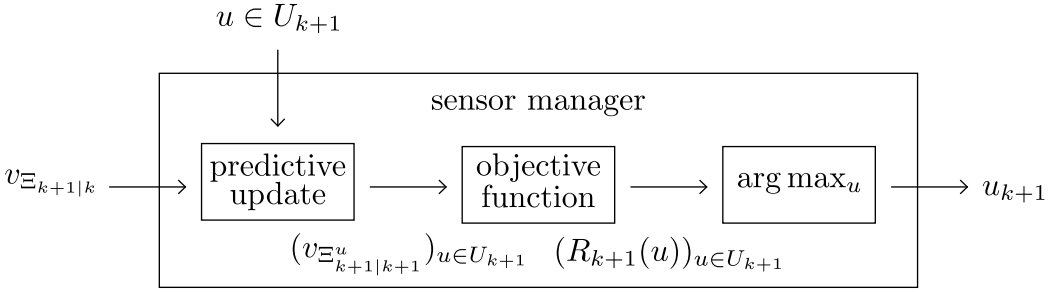
In this chapter, some simple RFSs were presented. The multi-target and multi-observation RFSs involved in the RFS filter equations must be reduced to these simple RFFs (Poisson, cluster, Bernouilli) in order to produce tractable approximations and be able to design the PHD filter. The construction of the exact PHD filter - in the single-sensor as well as the multi-sensor case - was thoroughly described, and the data update equations were analyzed qualitatively. The data update equation of the multi-sensor case being exceedingly difficult to compute in the general case, a joint partitioning method of the state space and the sensors was presented in order to simplify the data update without approximation. Finally, the usual multi-sensor approximations were compared to the exact solution on a theoretical level. The ICA looked promising and should be studied further on simulated data.

CHAPTER 3

# Multi-sensor management within the PHD framework

PRevious work on the sensor management problem within the PHD framework remains, to the author's knowledge, scarce. To be sure, Mahler introduced in [Mahl 04] the PENT manager but, according to the author's opinion, it seems to be ill-adapted to a broad range of surveillance activities. Quite recently, Ristic et al. [Rist 10b, Rist 11a] worked on a more general RFS-based sensor manager, and their simulation results seem to reinforce the author's opinion regarding the PENT manager. In this chapter, the goal is to design a multi-sensor manager within the PHD framework whose data flow can be depicted as follows:

**Figure 3.1** Data flow of the sensor management process




As illustrated in figure 3.1, the structure of the sensor manager is based on three distinctive features:

1. the predictive update: compute the *predictive PHD* $v^u_{\Xi_{k+1|k}}$, i.e. the PHD of the posterior RFS should the sensors be controlled according to some $u \in U_{k+1}$;

2. the objective function: determine a *reward* $R^u_{k+1}$ for each predictive posterior $v^u_{\Xi_{k+1|k}}$;

3. the selection step: select the control with the highest reward.

The first part of this chapter (section 3.1) focuses on the target extraction process which will be necessary for the predictive update step (see figure 3.1). Since the PHD is a density, the PHD filter does not directly provides tracks as in more classical track-based filters; thus, one must *extract* information about eventual targets from the propagated density. Usual extraction processes seem to be mainly based on clustering techniques such as the k-mean algorithm [Clar 06]. However, Tobias et al. argued [Tobi 08] that better extraction techniques could be designed by removing a target's worth of *weight* from the PHD $\hat{N}$ times, where $\hat{N}$ is the extracted number of targets. Notably, this method is bound to produce better results in the extraction of close targets, where the k-means algorithm would typically extract a single target averaging the two true targets. The target extractor presented in this thesis follows closely the solution Tobias et al. proposed. It should be noted that Tang et al. described [Tang 11] an improved extraction method, combining traditional clustering techniques such as the k-means algorithm with the solution proposed by Tobias et al., but this came too late to the author's attention to be considered in this thesis.

The second part (section 3.2) deals with the predictive update step, built as an extension of Mahler's work on the PIMS [Mahl 04] to the multi-sensor case.

The last part (section 3.3) focuses on the design of a sensor manager. It covers the description of Mahler's PENT manager and its inadequacy to some scenarri in surveillance activities. Consequently, the last part of section 3.3 is devoted to the construction of another sensor manager.

## 3.1 Target extraction

Since RFSs are random variables on "large" spaces $\mathcal{F}(\mathcal{X})$ where no sum operator is defined, the traditional expectation:

$$\mathbb{E}[\Xi(\omega)] = \int_{\mathcal{F}(\mathcal{X})} p_\Xi(X) X \mu(dX) \tag{3.1}$$

has no mathemical sense even if the probability density $p_\Xi$ is properly defined. Thus, usual estimators such as the Maximum A Posteriori cannnot be applied on RFS. Rather, one should exploit the fact that, aside from its probability density, a RFS can be described by its cardinality distribution $\rho_\Xi$ and its family of spatial distributions $\{P^{(n)}_\Xi\}_{n\in\mathbb{N}}$.

In this thesis, a *target extractor* is defined as an estimator on RFSs that:

1. estimates an *extracted target number* $\hat{N}$ - an *integer* - based on the cardinality distribution $\rho_{\Xi}$;
2. estimates the target configuration $\hat{X} = \{x_1, ..., x_{\hat{N}}\}$, a *finite family* on $\mathcal{X}$, based on the spatial distribution $P_{\Xi}^{(\hat{N})}$.

Note that $\hat{X}$ is built as a family and not a set, because the extracted targets are (generally) ordered. Target extractors are considerably easier to design when restricted to Poisson RFSs, since they are characterized by their intensity or PHD (definition 2.1). In this case, a target extractor is an estimator on Poisson RFSs that:

1. estimates an *extracted target number* $\hat{N}$ - an *integer* - based on the Poisson parameter $v_{\Xi}[1]$;
2. estimates the target configuration $\hat{X} = (x_1, ..., x_{\hat{N}})$, a *finite family* on $\mathcal{X}$, based on the PHD $v_{\Xi}(.)$.

In the scope of this thesis, targets need to be extracted from PHDs of Poisson RFSs only, namely:

- predicted PHDs $v_{\Xi_{k+1|k}}(.|Z_{1:k})$;
- predictive PHDs $v^{u}_{\Xi_{k+1|k}}(.|Z_{1:k})$;
- posterior PHDs $v_{\Xi_{k+1|k+1}}(.|Z_{1:k+1})$.

For simplicity's sake, time subscripts and dependence on past measurements will be omitted in this section and $v_{\Xi}$ will denote indifferently one of the PHDs above. It will also be assumed that $v_{\Xi}$ is continuous on $\mathcal{X}$.

### 3.1.1 Highest peaks extractor

Recall from proposition 1.3 that, given the multi-target PHD $v_{\Xi}$:

- the estimated target number is $N = v_{\Xi}[1] = \int_{\mathcal{X}} v_{\Xi}(x)dx$;
- the targets are i.i.d. according to the normalized PHD, i.e. the probability distribution $\frac{v_{\Xi}(.)}{N}$.

Thus, a "naive" *highest peaks extractor* could be defined as follows:

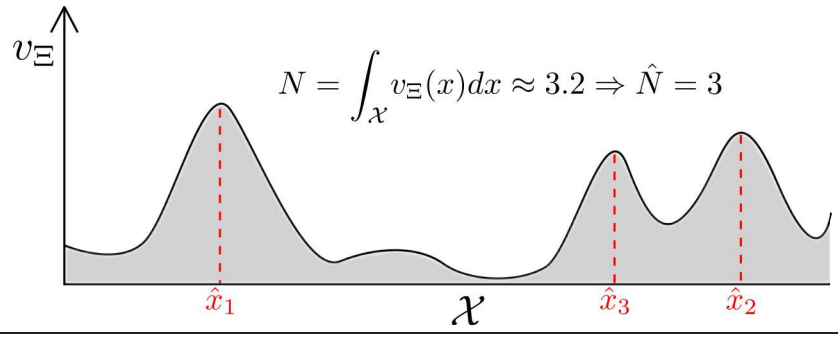
**Definition 3.1.** *For any PHD $v_\Xi$ on $\mathcal{X}$, the set of extracted targets from $v_\Xi$, if it exists, is the collection $\hat{X}^{HPE}(v_\Xi) \in \mathcal{X}$ defined by:*

$$\hat{X}^{HPE}(v_\Xi) \stackrel{def}{=} (\hat{x}_1, ..., \hat{x}_{\hat{N}}) \tag{3.2}$$

*where:*

- $\hat{N} \stackrel{def}{=} [v_\Xi[1]]_{nearest\ integer}$ *is the extracted number of targets;*
- $\forall n \in [1\ \hat{N}]$, $\hat{x}_n$ *is the n-th highest local extremum of the PHD $v_\Xi$.*

**Figure 3.2** Illustration of the HPE

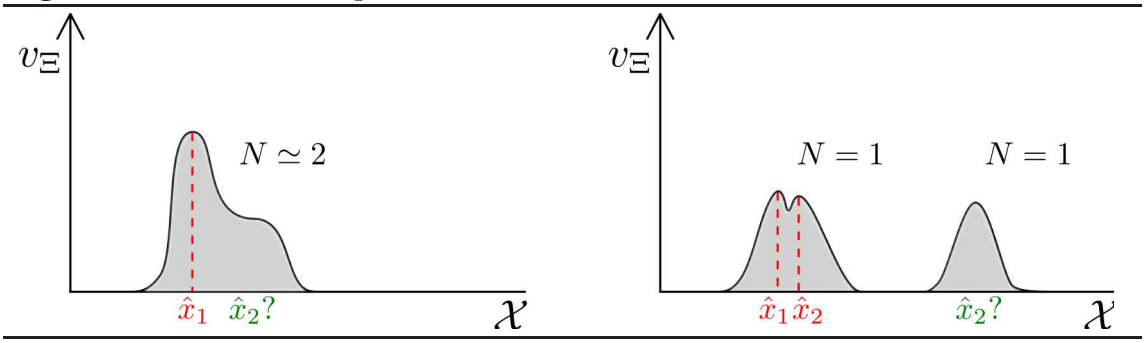



### 3.1.2 Weighted peaks extractor

Although the HPE is remarkable by its simplicity, problems may arise in specific situations such as shown in the following example:

**Example 3.1.** *Consider the following situations:*

**Figure 3.3** Improper target extraction by the HPE




*On the left hand side, it seems that two targets are close enough in the state space. HPE extracts target $\hat{x}_1$ at the first peak, but cannot extract a second target because*

*there are no more peaks. However, a second target is probably in the vicinity of* $\hat{x}_1$.

*On the right hand side, the HPE extracts the correct target number but the second target is extracted in the vicinity of* $\hat{x}_1$ *while the shape of the PHD suggests that the second target should be extracted around the third highest peak.*

Example 3.1 shows situations were the HPE does not performs as might be expected when target are getting closer (left figure) or when there are more than one peak that are likely to account for the same target (right figure). It seems that the HPE may fail because it does not really exploit the local distribution of the PHD. The following weight-based approach closely follows the solution proposed by Tobias et al. [Tobi 08] in which targets are extracted in regions whose weight (i.e., the integral of the PHD) reaches a given *target weight*:

**Definition 3.2.** *For any PHD* $v_\Xi$ *on* $\mathcal{X}$*, define:*

- *the extracted target number as* $\hat{N} \overset{def}{=} [v_\Xi[1]]_{nearest\ integer}$*;*
- *the target weight as* $W_t \overset{def}{=} \begin{cases} 0 & (\hat{N} = 0) \\ \dfrac{v_\Xi[1]}{\hat{N}} & (otherwise) \end{cases}$.

*Besides, for any positive function* $f(.)$ *on* $\mathcal{X}$*, any (strictly) positive real number* $r$*, any state point* $x_0 \in \mathcal{X}$*, defined as:*

- $\mathcal{B}_r(x_0) \overset{def}{=} \{x \in \mathcal{X} \mid d_\mathcal{X}(x_0, x) \leqslant r\}$ *the closed ball centered on* $x_0$ *with radius* $r$*;*
- $W(., d, f) \overset{def}{=} \int_{\mathcal{B}_d(.)} f(x)dx$ *the neighborhood weight function.*

*Initialize the weight function with the PHD, i.e.* $w^{(1)} = v_\Xi$*, and proceed as follows:*

- *Find new global maximum:* $x^{(n)} = \arg\max_x w^{(n)}(x)$*;*
- *Find new neighborhood span:* $d^{(n)} = \arg\min_d W(x^{(n)}, d, v_\Xi^{(n)}) \geqslant W_t$*;*
- *Set new neighborhood:* $N^{(n)} = \mathcal{B}_{d^{(n)}}(x^{(n)})$*;*
- *Set new neighborhood weight:* $W^{(n)} = W(x^{(n)}, d^{(n)}, w^{(n)})$*;*
- *Compute new target state:* $\hat{x}_n = \frac{\int_{N^{(n)}} x\ w_\Xi^{(n)}(x)dx}{W^{(n)}}$*;*
- *Compute new weight function* $w^{(n+1)}(.) = w^{(n)}(.) - 1_{N^{(n)}}(.)\frac{W_t}{W^{(n)}}$.

*The resulting family* $\hat{X}^{WE}(v_\Xi) \overset{def}{=} (\hat{x}_1, ..., \hat{x}_{\hat{N}})$*, if it exists, is the family of extracted targets.*

For simplicity's sake, $\hat{X}^{WE}(v_\Xi)$ will be shortened to $\hat{X}^{WE}$ when there is no ambiguity. Intuitively, the WE computes the smallest neighborhood around the highest point of the PHD with weight $W_t$, extracts the first target as the weighted average of the neighborhood, removes the neighborhood weight from the PHD and proceeds with the next extraction. Note that the target weight accounts for the common discrepancy between the expected number of targets $v_\Xi|1]$, which is *not* an integer in general, and the expected number of targets $\hat{N}$, an integer by construction. For example, if $v_\Xi|1] = 2.2$, then $\hat{N} = 2$ and the target weight is $W_t = \frac{2.2}{2} = 1.1$, such that $\hat{N}W_t = v_\Xi|1]$. Since the WE removes exactly $W_t$ of weight at each iteration (see definition 3.2), $\int_\mathcal{X} w^{(n)}(x)dx > 0$, $W^{(n)} > 0$ and thus the new target states $\hat{x}_n$ are well-defined.

**Example 3.2.** *Continuing example 3.1, the WE can be illustrated as follows:*

**Figure 3.4** Target extraction by the WE (1)

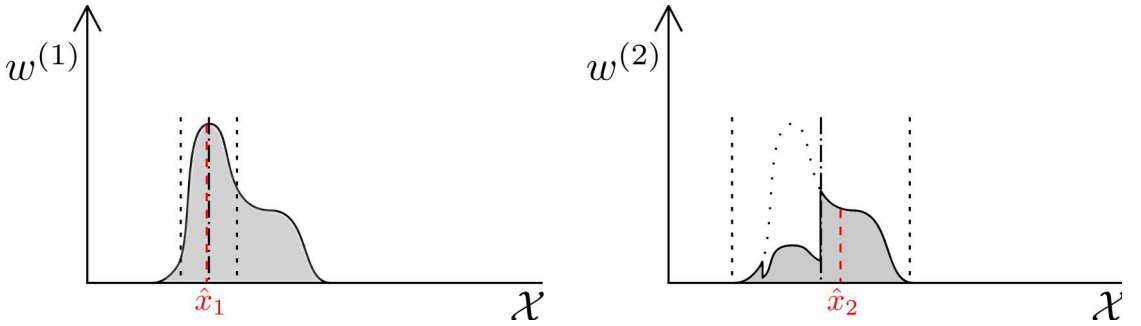
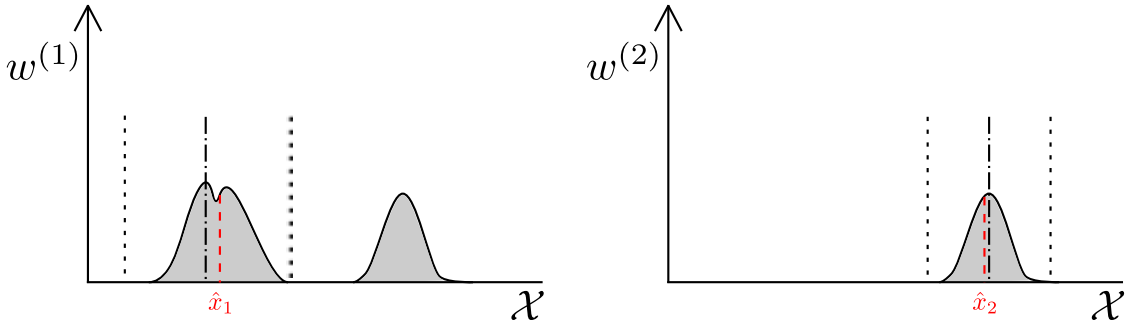



**Figure 3.5** Target extraction by the WE (2)




It is important to note that the WE has its own issues. While the finite *family* $\hat{X}^{WE}(v_\Xi) = (\hat{x}_n)_{n=[1\ \hat{N}]}$ is well-defined, the finite *set* $\hat{X}^{WE}(v_\Xi) = \bigcup_{n=1}^{\hat{N}} \hat{x}_n$ may not if some states are *identical*. However, this event being highly unlikely, it will be assumed from now on that the set of extracted targets exists, regardless of the initial weight distribution. Another issue is the dermination of the neighorhood

span, particularly for the last extracted target. By construction, $\int_{\mathcal{X}} w^{(\hat{N})}(x)dw = W_t$, which means that the last neighborhood may span to the entire state space, in which case setting the last target as the the weighted average may be a poor choice. In the practical implementation of the WE (see chapter 4), the span of the neighborhoods will be capped by a parameter.

## 3.2 Multi-sensor predictive PHD

In the scope of the thesis, multi-sensor controls have limited influence on the observation process. A control $u \in U_k$ aims at shaping the FOVs configuration, but has no effect on the measurement and false alarm process. That is, a control only shapes the detection probabilities $(p_k^{d,j}(.))_{j\in[1\ S]}$:

**Notation 3.1.** *If the sensors are under control $u \in U_k$, the following notations are used:*

$$(F_u^j)_{j\in[1\ S]} \overset{not}{=} (F_k^j)_{j\in[1\ S]} \tag{3.3}$$

$$(p_u^{d,j}(.))_{j\in[1\ S]} \overset{not}{=} (p_k^{d,j}(.))_{j\in[1\ S]} \tag{3.4}$$

### 3.2.1 Predictive update equation

The aim of the predictive update step for control $u$ is to guess, without new measurements $Z_{k+1}$, the shape of the *predictive RFS* $\Xi_{k+1|k+1}^u$, that is the expected posterior PHD should the sensors be under control $u$. In the PHD framework, the challenge is to compute the predictive PHD $v_{\Xi_{k+1|k+1}^u}$ based on the predicted PHD $v_{\Xi_{k+1|k}}$:

**Definition 3.3.** *At any time step $k+1$ and for any control $u \in U_{k+1}$, the predictive PHD $v_{\Xi_{k+1|k+1}^u}$ is given by:*

$$v_{\Xi_{k+1|k+1}^u}(.|Z_{1:k}) \overset{def}{=} \mathbb{E}[v_{\Xi_{k+1|k+1}}(.|Z_{1:k} \cup \Sigma_u(\omega))] \tag{3.5}$$

*where:*

- *$\Sigma_u$ is a predictive multi-sensor observation RFS (yet to be defined);*
- *$v_{\Xi_{k+1|k+1}}(.|Z_{1:k} \cup \Sigma_u(\omega))$ is the posterior multi-target RFS given by theorem 2.3.*

In definition 3.3, $\Sigma_u$ describes the multi-sensor observation process based on the known information about the current living targets $X_{k+1}$, i.e. based on the predicted PHD $v_{\Xi_{k+1|k}}$. Because the multi-sensor data update equation - either exact ((2.44) or (2.53)) or approximated ((2.65), (2.69) or (2.73)) - requires a multi-sensor measurement set in input, definition 3.3 is unexploitable unless the number of possible

realizations of RFS $\Sigma_u$ is *finite* - and, if possible, small enough - *and* its probability distribution $p_{\Sigma_u}(.)$ is known *explicitly*. In the RFS framework, this probability distribution is given by:

$$p_{\Sigma_u}(.) = \int_{\mathcal{F}(\mathcal{X})} p_{\Xi_{k+1|k}}(X) p_{\Sigma_u(X)}(.) \mu(dX) \tag{3.6}$$

where $\Sigma_u(X)$ is the predictive observation RFS conditionally on target set $X$. Of course, equation (3.6) is widely impractical since:

- according to the multi-sensor observation model (see proposition 2.4), for a given set $X$, $\Sigma_u(X)$ is a complicated RFS involving missed detections, false alarms and noise measurement processes;
- the set integral precludes a tractable implementation of $p_{\Sigma_u}$.

First of all, one must "discretize" RFS $\Sigma_u(X)$, i.e. provide an approximation with a *finite* number of realizations. In the general case, the single-sensor observation spaces $\mathcal{Z}^j$ are uncountable and thus the single-sensor observation RFSs $\Sigma^j_{k+1}(X)$ have values in uncountable spaces $\mathcal{F}(\mathcal{Z}^j)$. The first step is to discretize the observation spaces (adapted from [Mahl 04]):

**Definition 3.4.** *At any time step $k$, for any sensor $j$, the (ideal) measurement function $\rho^j_k(.)$ is the mapping defined as:*

$$\begin{array}{rccc} \rho^j_k : & \mathcal{X} & \to & \mathcal{Z}^j \\ & x & \mapsto & z = \rho^j_k(x) \end{array} \tag{3.7}$$

*where $z$ is the noiseless measurement produced by sensor $j$ from a detected target with state $x$.*

*For any finite set $X = \{x_1, ..., x_N\} \subset \mathcal{X}$, assuming that the functions $\rho^j_k(.)$ are injective, the multi-sensor ideal measurement set $Z^{Id}_k$ is the disjoint union:*

$$Z^{Id}_k(X) \overset{def}{=} \bigsqcup_{j=1}^{S} Z^{Id,j}_k(X) = \bigsqcup_{j=1}^{S} \rho^j_k(X) \tag{3.8}$$

*where $Z^{Id,j}_k(X) = \rho^j_k(X)$ is the single-sensor ideal measurement set of sensor $j$.*

Note that the injectivity of the measurement functions is required for the ideal measurement sets to be well-defined, even though this will not be the case in practical implementations (see chapter 4). In this chapter, the measurement functions will be considered bijective. The (multi-sensor) predictive observation RFS $\Sigma^u_{k+1}(X)$ can then be constructed as follows:

**Proposition 3.1.** *For any finite set $X = \{x_1, ..., x_N\} \subset \mathcal{X}$, any sensor $j$, any control $u \in U_{k+1}$, the (single-sensor) predictive observation RFS of sensor $j$, conditionally on target set $X$, is the RFS $\Sigma_u^j(X)$ with values in $\mathcal{F}(Z_{k+1}^{Id,j}(X))$ characterized by its probability density:*

$$p_{\Sigma_u^j(X)}(Z) \stackrel{def}{=} \prod_{z \in Z} \left(p_u^{d,j}((\rho_{k+1}^j)^{-1}(z))\right) \prod_{z \in Z_{k+1}^{Id,j}(X) \setminus Z} \left(1 - p_u^{d,j}((\rho_{k+1}^j)^{-1}(z))\right) \tag{3.9}$$

*where $Z_{k+1}^{Id,j}(X)$ is the ideal measurement set given by definition 3.4.*

*The predictive observation RFS, conditionally on target set $X$, is the joint RFS:*

$$\Sigma_u(X) \stackrel{def}{=} \bigsqcup_{j=1}^{S} \Sigma_u^j(X) \tag{3.10}$$

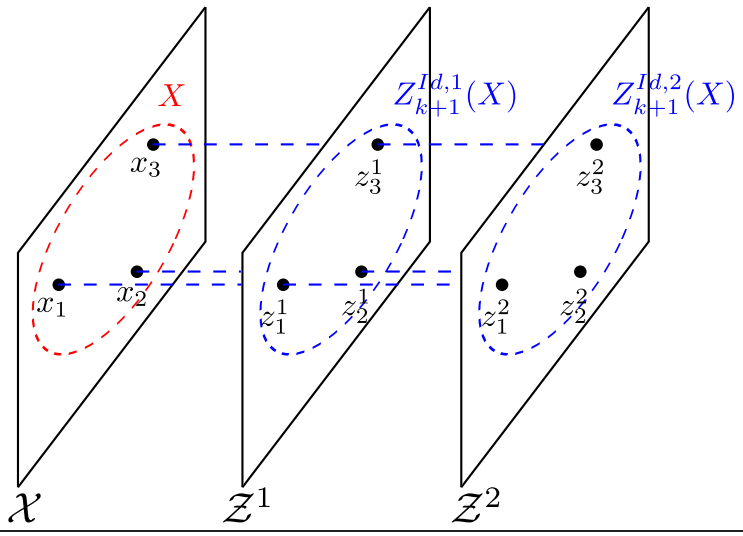
*Then, the probability distribution of the predictive observation RFS exists and, for any set $Z = \bigsqcup_{j=1}^{S} Z^j \in \bigsqcup_{j=1}^{S} \mathcal{F}(Z_{k+1}^{Id,j}(X))$:*

$$p_{\Sigma_u(X)}(Z) = \prod_{j=1}^{S} p_{\Sigma_u^j(X)}(Z^j) \tag{3.11}$$

The proof is straightforward using the property of joint RFSs (1.8).

**Example 3.3.** *Consider the multi-target set $X = \{x_1, x_2, x_3\}$ and the ideal measurement sets illustrated as follows:*

**Figure 3.6** Example of ideal measurement sets




*In addition, let $u$ be an available multi-sensor control such that:*

- $p_u^{d,1}(x_1) = 0$, $p_u^{d,1}(x_2) = \frac{2}{3}$, $p_u^{d,1}(x_3) = \frac{2}{3}$;
- $p_u^{d,2}(x_1) = \frac{1}{3}$, $p_u^{d,2}(x_2) = \frac{1}{2}$, $p_u^{d,2}(x_3) = \frac{2}{3}$.

*Then, the probability distribution of the predictive RFS $\Sigma_u^1(X)$ is given by:*

$$\begin{cases} p_{\Sigma_u^1(X)}(\emptyset) = \dfrac{1}{9} \\ p_{\Sigma_u^1(X)}(\{z_2^1\}) = p_{\Sigma_u^1(X)}(\{z_3^1\}) = \dfrac{2}{9} \\ p_{\Sigma_u^1(X)}(\{z_2^1, z_3^1\}) = \dfrac{4}{9} \end{cases} \tag{3.12}$$

*Likewise, the probability distribution of the predictive RFS $\Sigma_u^2(X)$ is given by:*

$$\begin{cases} p_{\Sigma_u^2(X)}(\{z_1^2\}) = p_{\Sigma_u^2(X)}(\{z_1^2, z_2^2\}) = \dfrac{1}{18} \\ p_{\Sigma_u^2(X)}(\emptyset) = p_{\Sigma_u^2(X)}(\{z_2^2\}) = \dfrac{1}{9} \\ p_{\Sigma_u^2(X)}(\{z_3^2\}) = p_{\Sigma_u^2(X)}(\{z_2^2, z_3^2\}) = \dfrac{2}{9} \\ p_{\Sigma_u^2(X)}(\{z_1^2, z_3^2\}) = p_{\Sigma_u^2(X)}(\{z_1^2, z_2^2, z_3^2\}) = \dfrac{1}{9} \end{cases} \tag{3.13}$$

As illustrated by example 3.3, the predictive RFS $\Sigma_u(X)$ covers all the possible measurements configuration - conditionally on the target set $X$ - provided that, for each sensor:

- the measurement process is noiseless;
- there are no false alarms.

However, the predictive RFS *does take into account* the missed detections.

Now that the predictive RFS $\Sigma_u(X)$ is properly defined and that its probability distribution is known explicitly thanks to proposition 3.1, the set integral in (3.6) is bypassed by considering a unique target set (inspired by [Mahl 04]):

**Definition 3.5.** *At any time step $k+1$, $Z_{k+1}^{WE}$ is the particular ideal measurement set $Z_{k+1}^{Id}(X)$ given by:*

$$Z_{k+1}^{WE} \stackrel{def}{=} Z_{k+1}^{Id}\left(\chi(\hat{X}^{WE})\right) \tag{3.14}$$

*Likewise, for any control $u \in U_{k+1}$, the PIMS $\Sigma_u^{WE}$ is the particular predictive RFS $\Sigma_u(X)$ given by:*

$$\Sigma_u^{WE} \stackrel{def}{=} \Sigma_u\left(\chi(\hat{X}^{WE})\right) \tag{3.15}$$

*that is, $\Sigma_u^{WE}$ is an approximation of $\Sigma_u$ characterized by the probability distribution:*

$$p_{\Sigma_u^{WE}}(.) = p_{\Sigma_u\left(\chi(\hat{X}^{WE})\right)}(.) \simeq \int_{\mathcal{F}(\mathcal{X})} p_{\Xi_{k+1|k}}(X) p_{\Sigma_u(X)}(.)\mu(dX) = p_{\Sigma_u}(.) \tag{3.16}$$

Note that definition 3.5 could be easily extended to other target extractors, leading to different PIMSs - e.g. $\Sigma_u^{HPE}$.

Thanks to proposition 3.1 and definition 3.5, the computation of the predictive PHD (3.5) is tractable and, combined with previous result (2.53), yields:

**Proposition 3.2.** *Under the assumptions given in theorem 2.4, for any control $u \in U_{k+1}$, the predictive PHD $v_{\Xi^u_{k+1|k+1}}(.|Z_{1:k})$ is given by:*

$$\begin{aligned} & v_{\Xi^u_{k+1|k+1}}(x|Z_{1:k}) \\ & = \beta[\delta_\emptyset, \delta_x]K_{\mathcal{X}}^{-1} + \sum_{Z \subseteq Z_{k+1}^{WE}} p_{\Sigma_u^{WE}}(Z) \frac{\displaystyle\sum_{C \in \mathcal{C}(Z)} \sum_{C_i \in C} \left( \beta[\delta_{C_i}, \delta_x] \prod_{C_j \neq C_j} \beta[\delta_{C_j}, 1] \right)}{\displaystyle\sum_{C \in \mathcal{C}(Z)} \prod_{C_i \in C} \beta[\delta_{C_i}, 1]} K_{\mathcal{X}}^{-1} \end{aligned} \tag{3.17}$$

*or, equivalently:*

$$v^u_{\Xi_{k+1|k+1}}(x|Z_{1:k}) = \sum_{Z \subseteq Z_{k+1}^{WE}} p_{\Sigma_u^{WE}}(Z) \frac{\left[\frac{\delta}{\delta x}\left(\frac{\delta}{\delta Z} e^{\beta[\delta_\emptyset, \bar{g}, h]}\right)\right]_{g^{1\ldots S}=0, h=1}}{\left[\frac{\delta}{\delta Z} e^{\beta[\delta_\emptyset, \bar{g}, h]}\right]_{g^{1\ldots S}=0, h=1}} K_{\mathcal{X}}^{-1} \tag{3.18}$$

*where the ideal measurement set $Z_{k+1}^{WE}$ and the PIMS $\Sigma_u^{WE}$ are given by definition 3.5.*

The proof is given in appendix A. Note that the cross-terms in equations (3.17) and (3.18) implictly depend on the control $u$ through the detection probability functions $p_u^{d,j}(.)$ (see notation (3.4)).

As expected, proposition 3.2 is simplified in the single-sensor case and yields Mahler's result [Mahl 04]:

**Corollary 3.1.** *Under the assumptions given in theorem 2.2, for any control $u \in U_{k+1}$, the predictive PHD $v_{\Xi^u_{k+1|k+1}}(.|Z_{1:k})$ is given by:*

$$v_{\Xi^u_{k+1|k+1}}(.|Z_{1:k}) = \left(1 - p_u^d(.) + \sum_{z \in Z^{WE}_{k+1}} p_u^d((\rho_{k+1})^{-1}(z)) \frac{p_u^d(.) L^z_{k+1}(.)}{\lambda^c_{k+1} c_{k+1}(z) + v_{\Xi_{k+1|k}}[p_u^d L^z_{k+1}]}\right) v_{\Xi_{k+1|k}}(.|Z_{1:k}) \tag{3.19}$$

*where the ideal measurement set $Z^{WE}_{k+1}$ is given by definition 3.5.*

The proof is given in appendix A.

### 3.2.2 Simplification by joint partitioning

Result (3.19) is instructive because it shows that in the single-sensor case, the costs for the computation of the predictive PHD $v_{\Xi^u_{k+1|k+1}}$ and the posterior PHD $v_{\Xi_{k+1|k+1}}$ (equation (2.29)) are similar - linear respective to the measurement number. The construction of the predictive single-sensor PHD is quite intuitive. The current measurements being decoupled in the single-sensor case, the sum in equation (2.29) can be seen as the unweighted sum of the inlfuence of each produced measurement on the posterior PHD. The construction of the predictive PHD is similar, except that no current measurements have been produced yet and one must weight the influence of each ideal measurement with its probability of occurence, which is exactly the probability of detection of the associated target.

Unfortunately, this interpretation does not hold in the general case, because the effect of each current measurement cannot be isolated in (2.53). Thus, the predictive PHD equation (3.17) requires the computation of an exact multi-sensor data update step (2.53) *for every possible subset $Z$* of the ideal measurement set $Z^{WE}_{k+1}$. However, proposition 3.2 can be significantly simplified *without approximation* by using the previous joint partitioning (see chapter 2). More precisely, the ideal measurement set $Z^{WE}_{k+1}$ can be partitioned as follows:

**Definition 3.6.** *For any control $u \in U_{k+1}$, let $(S_u(p))_{p=1}^{P_u}, (T_u(p))_{p=0}^{P_u}$ be the joint partition as given by definition 2.9. Then, the set of extracted targets $\hat{X}^{WE}$ can be partitioned as follows:*

$$\forall p \in [0\ P_u],\ \hat{X}^{WE}_{u,p} \overset{def}{=} \hat{X}^{WE} \cap T_u(p) \tag{3.20}$$

*Then, the ideal measurement set $Z^{WE}_{k+1}$ can be reduced as follows:*

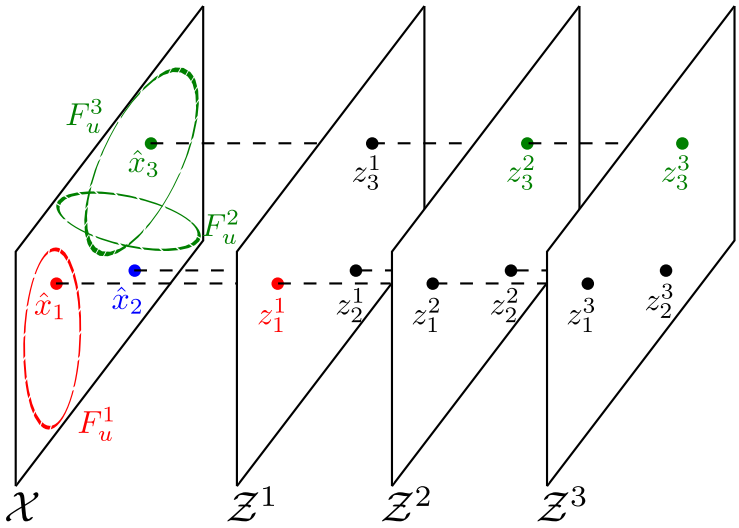
$$\forall p \in [1\ P_u],\ Z^{WE}_{u,p} \overset{def}{=} \bigsqcup_{j \in S_u(p)} Z^{WE,j}_{u,p} = \bigsqcup_{j \in S_u(p)} \rho^j_{k+1}(\hat{X}^{WE}_{u,p}) \tag{3.21}$$

*Finally, for any* $p \in [1\ P_u]$, $\Sigma_{u,p}^{WE}$ *is the reduction of the PIMS* $\Sigma_u^{WE}$ *to partition element* $p$, *i.e. the RFS with values in* $\mathcal{F}(Z_{u,p}^{WE})$ *such that, for any set* $Z = \bigsqcup_{j \in S_u(p)} Z^j \in \bigsqcup_{j \in S_u(p)} \mathcal{F}(Z_{u,p}^{WE,j})$:

$$p_{\Sigma_{u,p}^{WE}}(Z) \overset{def}{=} \prod_{j \in S_u(p)} \left( \prod_{z \in Z^j} \left(p_u^{d,j}((\rho_{k+1}^j)^{-1}(z))\right) \prod_{z \in Z_{u,p}^{WE,j} \setminus Z^j} \left(1 - p_u^{d,j}((\rho_{k+1}^j)^{-1}(z))\right) \right) \tag{3.22}$$

**Example 3.4.** *Consider the set of extracted targets* $\hat{X}^{WE} = \{\hat{x}_1, \hat{x}_2, \hat{x}_2\}$ *and the FOVs* $F_u^1$, $F_u^2$, $F_u^3$ *illustrated as follows:*

**Figure 3.7** Joint partitioning of $\hat{X}^{WE}$ and $Z_{k+1}^{WE}$ (p = 0 in blue, p = 1 in green, p = 2 in red)




*First of all, since* $F_u^2 \cap F_u^3 \neq \emptyset$, *the joint partitioning is, according to definition 2.9:*

- *sensors:* $S_u(1) = \{1\}$ *and* $S_u(2) = \{2, 3\}$;
- *state space:* $T_u(1) = F_u^1$, $T_u(2) = F_u^2 \cup F_u^3$ *and* $T_u(0) = \mathcal{X} \setminus (F_u^1 \cup F_u^2 \cup F_u^3)$.

*Then, the extracted targets* $\hat{X}^{WE}$ *are partitioned and the ideal measurements* $Z_{k+1}^{WE}$ *are reduced according to definition 3.6:*

- *extracted targets:* $\hat{X}_{u,0}^{WE} = \{\hat{x}_2\}$, $\hat{X}_{u,1}^{WE} = \{\hat{x}_1\}$ *and* $\hat{X}_{u,2}^{WE} = \{\hat{x}_3\}$;
- *ideal measurements:* $Z_{u,1}^{WE} = \{z_1^1\}$, *and* $Z_{u,2}^{WE} = \{z_3^2, z_3^3\}$.

*Note that ideal measurement $z_1^2$ is discarded because, under control $u$, $p_u^{d,2}((\rho_{k+1}^2)^{-1}(z_1^2)) = p_u^{d,2}(\hat{x}_1) = 0$ since $\hat{x}_1 \notin F_u^2$. Likewise, measurements $z_1^3$, $z_2^1$, $z_2^2$, $z_2^3$ and $z_3^1$ are discarded.*

With this joint partitioning, proposition 3.2 is simplified in a similar manner as theorem 2.4:

**Proposition 3.3.** *Under the assumptions given in proposition 3.2, for any control $u \in U_{k+1}$, the predictive PHD $v_{\Xi^u_{k+1|k+1}}(.|Z_{1:k})$ is given by:*

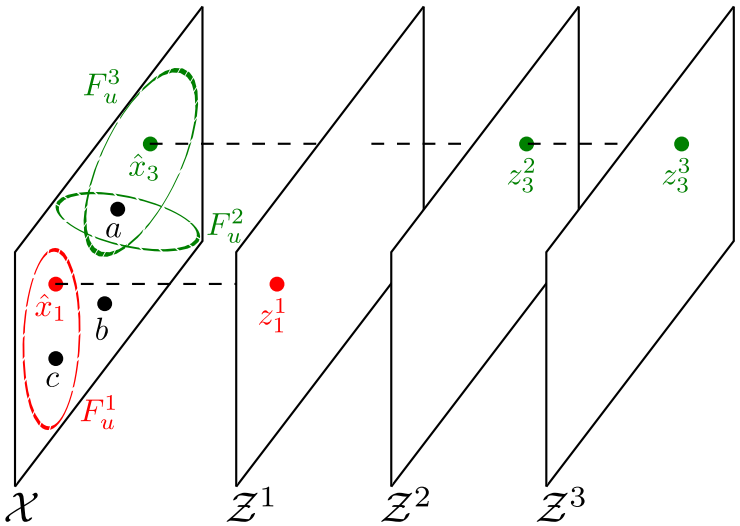
$$v_{\Xi^u_{k+1|k+1}}(x|Z_{1:k}) = \begin{cases} v_{\Xi_{k+1|k}}(x|Z_{1:k}) & (x \in T_u(0)) \\ \beta_p[\delta_\emptyset, \delta_x]K_{\mathcal{X}}^{-1} + \sum\limits_{Z \subseteq Z_{u,p}^{WE}} p_{\Sigma_{u,p}^{WE}}(Z) \dfrac{\sum\limits_{C \in \mathcal{C}(Z)} \sum\limits_{C_i \in C} \left( \beta_p[\delta_{C_i}, \delta_x] \prod\limits_{C_j \neq C_j} \beta_p[\delta_{C_j}, 1] \right)}{\sum\limits_{C \in \mathcal{C}(Z)} \prod\limits_{C_i \in C} \beta_p[\delta_{C_i}, 1]} K_{\mathcal{X}}^{-1} & (x \in T_u(p), p \neq 0) \end{cases} \tag{3.23}$$

*where $(Z_{u,p}^{WE})_{p=1}^{P_u}$ and $(\Sigma_{u,p}^{WE})_{p=1}^{P_u}$ are given by definition 3.6.*

The proof is given in appendix A.

**Example 3.5.** *Continuing with example 3.4, assuming that the predictive PHD must be computed in three target states a, b, c as shown in the followwing picture:*

**Figure 3.8** Computation of the predictive PHDs in points $a$, $b$, $c$



*Since $a \in T_u(2)$, using proposition 3.3 gives:*

$$v_{\Xi^u_{k+1|k+1}}(a|Z_{1:k}) = \beta_2[\delta_\emptyset, \delta_a]K_{\mathcal{X}}^{-1} + p_u^{d,2}(\hat{x}_3)(1-p_u^{d,3}(\hat{x}_3))\frac{\beta_2[\delta_{\{z_3^2\}}, \delta_a]}{\beta_2[\delta_{\{z_3^2\}}, 1]}K_{\mathcal{X}}^{-1}$$
$$+ (1-p_u^{d,2}(\hat{x}_3))p_u^{d,3}(\hat{x}_3)\frac{\beta_2[\delta_{\{z_3^3\}}, \delta_a]}{\beta_2[\delta_{\{z_3^3\}}, 1]}K_{\mathcal{X}}^{-1}$$
$$+ p_u^{d,2}(\hat{x}_3)p_u^{d,3}(\hat{x}_3)\frac{\beta_2[\delta_{\{z_3^2\}}, \delta_a]\beta_2[\delta_{\{z_3^3\}}, 1] + \beta_2[\delta_{\{z_3^2\}}, 1]\beta_2[\delta_{\{z_3^3\}}, \delta_a] + \beta_2[\delta_{\{z_3^2, z_3^3\}}, \delta_a]}{\beta_2[\delta_{\{z_3^2\}}, 1]\beta_2[\delta_{\{z_3^3\}}, 1] + \beta_2[\delta_{\{z_3^2, z_3^3\}}, 1]}K_{\mathcal{X}}^{-1}$$

*Then, since $b \in T_u(0)$:*

$$v_{\Xi^u_{k+1|k+1}}(b|Z_{1:k}) = v_{\Xi_{k+1|k}}(b|Z_{1:k})$$

*Finally, since $c \in T_u(1)$:*

$$v_{\Xi^u_{k+1|k+1}}(c|Z_{1:k}) = \beta_1[\delta_\emptyset, \delta_c]K_{\mathcal{X}}^{-1} + p_u^{d,1}(\hat{x}_1)\frac{\beta_1[\delta_{\{z_1^1\}}, \delta_c]}{\beta_1[\delta_{\{z_1^1\}}, 1]}K_{\mathcal{X}}^{-1}$$

*Note that, without the joint partitioning, using proposition 3.2 requires the computation of $2^{|\Sigma^{WE}_{k+1}|} = 2^9 = 512$ ratios for each point a, b, c.*

As usual, from proposition 3.3 immediately follows the equivalent derivative form:

**Corollary 3.2.** *Under the assumptions given in proposition 3.3, for any control $u \in U_{k+1}$, the predictive PHD $v_{\Xi^u_{k+1|k+1}}(.|Z_{1:k})$ is given by:*

$$v_{\Xi^u_{k+1|k+1}}(x|Z_{1:k})$$
$$= \begin{cases} v_{\Xi_{k+1|k}}(x|Z_{1:k}) & (x \in T_u(0)) \\ \displaystyle\sum_{Z \subseteq Z^{WE}_{u,p}} p_{\Sigma^{WE}_{u,p}}(Z)\frac{\left[\frac{\delta}{\delta x}\left(\frac{\delta}{\delta Z}e^{\beta_p[\delta_\emptyset, \bar{g}, h]}\right)\right]_{g^{j\in S_u(p)}=0, h=1}}{\left[\frac{\delta}{\delta Z}e^{\beta_p[\delta_\emptyset, \bar{g}, h]}\right]_{g^{j\in S_u(p)}=0, h=1}}K_{\mathcal{X}}^{-1} & (x \in T_u(p), p \neq 0) \end{cases} \tag{3.24}$$

*where $(Z^{WE}_{u,p})_{p=1}^{P_u}$ and $(\Sigma^{WE}_{u,p})_{p=1}^{P_u}$ are given by definition 3.6.*

### 3.2.3 A few leads for approximations

Arguably, using proposition 3.3 is still computationally intensive if the FOVs configuration is unfavorable. There are at least two leads, which can eventually be combined, mayt provide approximations with lighter computation costs.

First, since the construction of the predictive PHD is based on the data update equation (2.53), the approximations described in section 2.4 are still valid in this context. Of course, the resulting predictive PHDs will share the same limitations as the posterior PHD - e.g. the dependence on the sensor order if the ICA is used, although it may have less consequences in this case since the predictive PHD is used to evaluate a multi-sensor control rather than propagate the filtered state. Presumably, using an approximation such as the ICA would sometimes produce errors in the evaluation of the predictive PHDs large enough for the sensor manager to select a suboptimal control.

Another possible approximation of proposition 3.3 is to simplify the predictive RFSs $(\Sigma_{u,p}^{WE})_{p=1}^{P_u}$. Recall from proposition 3.1 that $\Sigma_u^{WE}$ is built as an approximation of the multi-sensor observation RFS in which the false alarms process and the noise in the measurement process of each sensor is discarded, but the missed detections are still considered. Thus, an easy way to further approximate the multi-observation RFS is to reduce the predictive RFS $\Sigma_u^{WE}$ to the following:

$$\forall p \in [1\ P_u],\ p_{\Sigma_{u,p}^{WE}}(Z) \overset{def}{=} \begin{cases} 1 & (Z = Z_{u,p}^{WE}) \\ 0 & (otherwise) \end{cases} \tag{3.25}$$

that is, to assume a full detection of the extracted targets. In this situation, the sum in proposition 3.3 and the predictive step reduces to a single data update step with the ideal measurement set $Z_{u,p}^{WE}$ as input.

## 3.3 Sensor manager

This section focuses on the design of a sensor manager and is divided in two parts. First, the PENT manager developed by Mahler is described and its inconsistency is shown on simple examples. Based on the flaws of the PENT, another approach is proposed for the first time - the BET manager.

### 3.3.1 The PENT manager

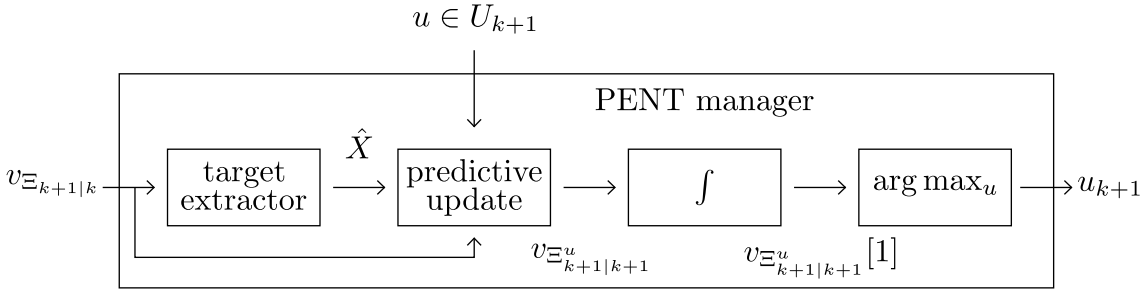
The PENT(-based) objective function was developed by Mahler following the PIMS construction, it is indeed a very simple objective function that naturally flows from the PIMS (adapted from [Mahl 04]):

**Definition 3.7.** *At any time $k+1$, the posterior expected target number (PENT) objective function is defined by:*

$$\begin{aligned} R_{k+1}:\quad & U_{k+1} \to \mathbb{R}^+ \\ & u \mapsto R_{k+1}(u) = v_{\Xi_{k+1|k+1}^u}[1] \end{aligned} \tag{3.26}$$

The data flow of the PENT manager is thus very simple and follows closely the generic flow given in figure 3.1:

**Figure 3.9** Data flow of the PENT manager




In other words, the PENT(-based) manager computes the predictive PHD for each possible control, and selects the control with the highest PENT. Definition 3.7, however, gives no insight on the shape of the FOVs configuration provided by the selected control. Proposition 3.3 does help a little on this point. Considering a control $u \in U_{k+1}$ with partitions $(T_u(p))_{p\in[0\ P_u]}$ and $(S_u(p))_{p\in[1\ P_u]}$:

1. If $x \in T_u(0)$, then $v_{\Xi^u_{k+1|k+1}}(x|Z_{1:k}) = v_{\Xi_{k+1|k}}(x|Z_{1:k})$. Thus, $v_{\Xi^u_{k+1|k+1}}[1_{T_u(0)}] = v_{\Xi_{k+1|k}}[1_{T_u(0)}]$: the expected target number in areas without any sensor coverage will remain identical.

2. If $x \in T_u(p), p \neq 0$, such that $Z^{WE}_{u,p} = \emptyset$, then $v_{\Xi^u_{k+1|k+1}}(x|Z_{1:k}) = \beta_p[\delta_\emptyset, \delta_x]K_{\mathcal{X}}^{-1} = \prod_{j\in S_u(p)} \left(1 - p_u^{d,j}(x)\right) v_{\Xi_{k+1|k}}(x|Z_{1:k})$. Thus, $v_{\Xi^u_{k+1|k+1}}[1_{T_u(p)}] \leqslant v_{\Xi_{k+1|k}}[1_{T_u(p)}]$: the expected target number in areas covered by at least one FOV but without any extracted target will decrease.

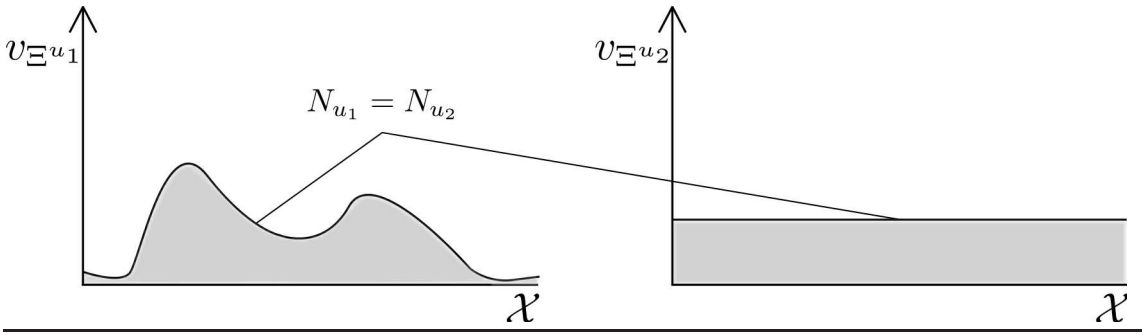
3. In partition elements $p$ with at least one associated ideal measurement ($Z^{WE}_{u,p} \neq \emptyset$) or, equivalently, at least one extracted target ($\hat{X}^{WE}_{u,p} \neq \emptyset$), the analysis on the contribution of each measurement on the posterior PHD in section 2.3.1 suggests that the expected target number will evolve toward the number of extracted targets $|\hat{X}^{WE}_{u,p}|$.

This quick analysis suggests that the PENT manager is unlikely to dissipate the sensors' effort in *unexplored* areas of the state space, but rather to focus on *previously detected* targets. Yet the PENT objective function is not entirely satisfying because it shares the same default as the HPE: the PENT being a *global* criteria which is based on the cardinality distribution of the predictive RFSs $\Xi^u_{k+1|k+1}$ only, the information given by the predictive PHDs on the target distribution is completely discarded. For this reason, Ristic et al. [Rist 11a] argued that the PENT manager

is expected to perform poorly in target localization problems and illustrated their point on simulated data. The following example shows some situations in which the PENT manager may appear ill-adapted for tracking purposes:

**Example 3.6.** *Consider the two following predictive PHDs, for available controls* $u_1, u_2 \in U_{k+1}$*:*

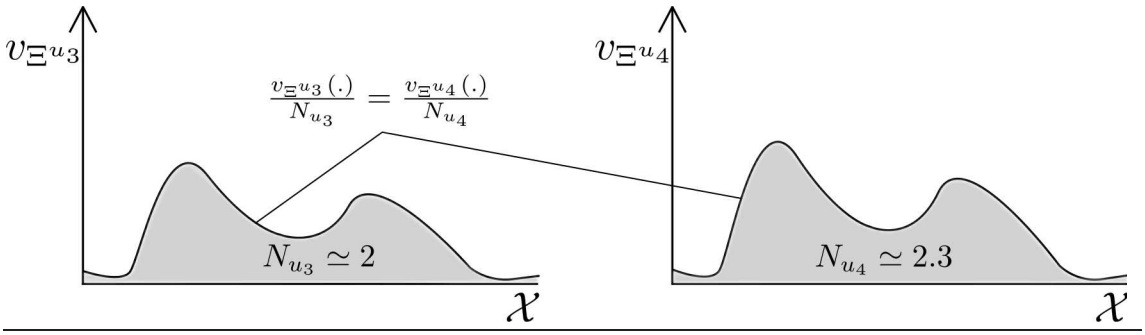
**Figure 3.10** Comparison of predictive PHDs (1)




*Controls* $u_1$ *and* $u_2$ *are equivalent regarding the PENT* ($N_{u_1} = N_{u_2}$)*, thus neither* $u_1$ *nor* $u_2$ *is favored by the PENT manager. However, control* $u_1$ *provides more information than* $u_2$ *regarding the localization of the estimated targets.*

*Then, consider the two following predictive PHDs, for available controls* $u_3, u_4 \in U_{k+1}$*:*

**Figure 3.11** Comparison of predictive PHDs (2)



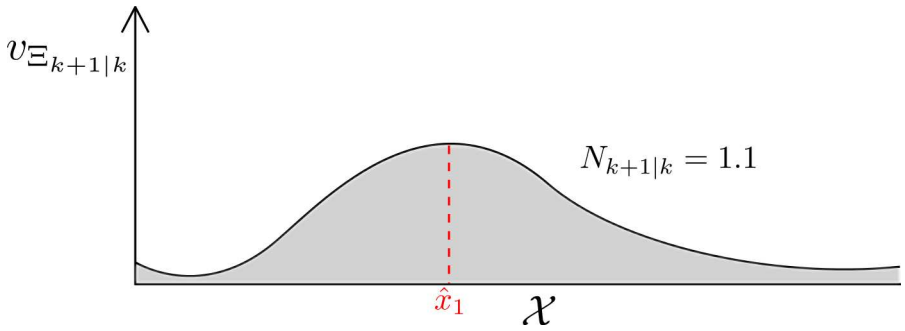

*Controls* $u_3$ *and* $u_4$ *are equivalent regarding the localization of expected targets* ($\frac{v_{\Xi^{u_3}}(.)}{N_{u_3}} = \frac{v_{\Xi^{u_4}}(.)}{N_{u_4}}$)*, but control* $u_4$ *is favored over control* $u_3$ *since its PENT is higher* ($N_{u_4} > N_{u_3}$)*. However, either the WE or the HPE would extract the same information from the two predictive PHDs.*

The examples above are instructive because they illustrate a situation in which the PENT objective function does not discriminate two controls while one may want to

do so (figure 3.10) and, conversely, a situation in which the PENT manager makes a decision that may appear irrelevant (figure 3.11). The next example is perhaps more suggestive, because it shows a simple situation in which the PENT manager makes the *wrong* decision:

**Example 3.7.** *Consider the following predicted PHD* $v_{\Xi_{k+1|k}}$ *at time* $k+1$*:*

**Figure 3.12** Predicted PHD $v_{\Xi_{k+1|k}}$




*That is, there is probably one target in the state space, whose state is estimated at* $\hat{x}_1$ *by the WPE. Assume that there is currently only one available sensor, producing true measurements only* ($\lambda^c_{k+1} = 0$) *and with no missed detection* ($p^d_{k+1} = 1$ *inside the FOV). Further assume that two controls are available:*

- *control* $u_1$*: the FOV covers all the state space, i.e.* $p^d_{u_1}(.) = 1$*;*
- *control* $u_2$*: the sensor is "shut down", i.e.* $p^d_{u_2}(.) = 0$*.*

*The ideal measurement set* $Z^{WE}_{k+1}$ *being reduced to a single element* $z_1 = \rho_{k+1}(\hat{x}_1)$*, computing the predictive PHD is straightforward in both cases with corollary 3.1:*

$$\begin{aligned} & v_{\Xi^{u_1}_{k+1|k+1}}(.|Z_{1:k}) \\ &= \left( 1 - \underbrace{p^d_{u_1}(.)}_{=1} + \underbrace{p^d_{u_1}(\hat{x}_1)}_{=1} \frac{\overbrace{p^d_{u_1}(.)}^{=1} L^{z_1}_{k+1}(.)}{\underbrace{\lambda^c_{k+1} c_{k+1}(z_1)}_{=0} + v_{\Xi_{k+1|k}}[\underbrace{p^d_{u_1}}_{=1} L^{z_1}_{k+1}]} \right) v_{\Xi_{k+1|k}}(.|Z_{1:k}) \\ &= \frac{L^{z_1}_{k+1}(.) v_{\Xi_{k+1|k}}(.|Z_{1:k})}{v_{\Xi_{k+1|k}}[L^{z_1}_{k+1}]} \end{aligned} \tag{3.27}$$

*Since* $p^d_{u_2}(\hat{x}_1) = 0$, $v_{\Xi^{u_2}_{k+1|k+1}}(.|Z_{1:k})$ *has an even simpler expression:*

$$\begin{aligned} & v_{\Xi^{u_2}_{k+1|k+1}}(.|Z_{1:k}) \\ &= \left(1 - \underbrace{p^d_{u_2}(.)}_{=0} + \underbrace{p^d_{u_2}(\hat{x}_1)}_{=0} \frac{p^d_{u_2}(.) L^{z_1}_{k+1}(.)}{\lambda^c_{k+1} c_{k+1}(z_1) + v_{\Xi_{k+1|k}}[p^d_{u_2} L^{z_1}_{k+1}]}\right) v_{\Xi_{k+1|k}}(.|Z_{1:k}) \\ &= v_{\Xi_{k+1|k}}(.|Z_{1:k}) \end{aligned} \tag{3.28}$$

*Then, the PENT objective function gives:*

$$\begin{cases} R_{k+1}(u_1) = v_{\Xi^{u_1}_{k+1|k+1}}[1] = \dfrac{v_{\Xi_{k+1|k}}[L^{z_1}_{k+1}]}{v_{\Xi_{k+1|k}}[L^{z_1}_{k+1}]} = 1 \\ R_{k+1}(u_2) = v_{\Xi^{u_2}_{k+1|k+1}}[1] = v_{\Xi_{k+1|k}}[1] = N_{k+1|k} = 1.1 \end{cases} \tag{3.29}$$

*Thus, regardless of the sensor measurement accuracy (that is, the shape of the likelihood function* $L_{k+1}$*), the PENT manager selects control* $u_2$ *and shuts down the sensor for the current time step.*

*Let* $X_{k+1} = \{x_1, ..., x_N\}$ *be the true multi-target set. Since there is no false alarm and no missed detections, should* $u_1$ *be selected, the current measurement set* $Z_{k+1}$ *would nessarily have the same size* $N$ *and equation* (2.29) *yields:*

$$\begin{aligned} & v_{\Xi_{k+1|k+1}}(.|Z_{1:k+1}) \\ &= \left(1 - \underbrace{p^d_{u_1}(x)}_{=1} + \sum_{z \in Z_{k+1}} \frac{\overbrace{p^d_{u_1}(.)}^{=1} L^z_{k+1}(.)}{\underbrace{\lambda^c_{k+1} c_{k+1}(z)}_{=0} + v_{\Xi_{k+1|k}}[\underbrace{p^d_{u_1}}_{=1} L^z_{k+1}]}\right) v_{\Xi_{k+1|k}}(.|Z_{1:k}) \\ &= \sum_{z \in Z_{k+1}} \frac{L^z_{k+1}(.) v_{\Xi_{k+1|k}}(.|Z_{1:k})}{v_{\Xi_{k+1|k}}[L^z_{k+1}]} \end{aligned}$$

*Thus:*

$$N_{k+1|k+1} = v_{\Xi_{k+1|k+1}}[1] = \sum_{z \in Z_{k+1}} \frac{v_{\Xi_{k+1|k}}[L^z_{k+1}]}{v_{\Xi_{k+1|k}}[L^z_{k+1}]} = |Z_{k+1}| = N$$

*Likewise, should* $u_2$ *be selected, the current measurement set* $Z_{k+1}$ *would necessarily be empty and thus:*

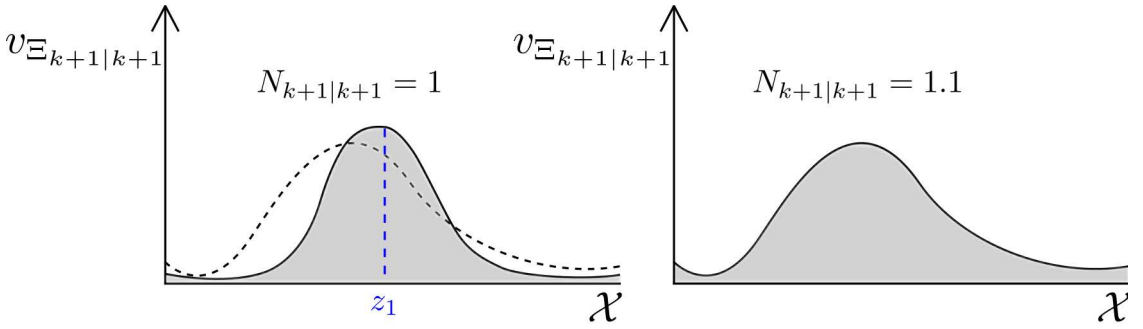
$$v_{\Xi_{k+1|k+1}}(.|Z_{1:k+1}) = \left(1 - \underbrace{p^d_{u_2}(x)}_{=0}\right) v_{\Xi_{k+1|k}}(.|Z_{1:k}) = v_{\Xi_{k+1|k}}(.|Z_{1:k})$$

*Thus:*

$$N_{k+1|k+1} = v_{\Xi_{k+1|k+1}}[1] = v_{\Xi_{k+1|k}}[1] = N_{k+1|k} = 1.1$$

*That is, control $u_1$ will necessarily provide a better estimation of the target number than $u_2$. Moreover, depending on the likelihood function, $u_1$ is likely to provide a better estimation of the target distribution as well, as illustrated in the following figures:*

**Figure 3.13** Posterior PHDs $v_{\Xi_{k+1|k+1}}$ (single true target)




a) Control $u_1$ b) Control $u_2$

*In conclusion, the PENT manager will select control $u_2$ even if control $u_1$ provides:*

- *a better estimation of the target number, with certainty;*
- *a better estimation of the target distribution, with high probability.*

To be sure, these examples do not prove that the PENT is unfit in every situations; actually, it has been used in previous works. Wei et al. [Wei 08a, Wei 08b] designed an interesting two-level sensors architecture in which the date update step is *synchronised* for the sensors belonging to a given *cluster*, but *sequential* between each cluster. The main filter is track-based, but the PHD formulation and the PENT objective function are implemented in a predictive step designed to select the order in which the clusters are to be processed at each data update step. In other works [El F 08, Zate 08], a PHD-based filter is implemented for a space object tracker and the sensor manager is built on the *posterior expected number of targets of interest* (PENTI), an extension of the PENT [Mahl 07c] in which the state points are weighted according to an interest function (i.e. the value of the PHD in all the state space does not contribute equally to the PENT).

Note that the PENT manager is specifically designed for controls shaping the FOV configuration, because the value of the PENT is much more sensible to the variation of the number of ideal measurements than, say, the variation in the shape of

the likelihood functions (see the analysis on the contribution of measurements on the posterior PHD in section 2.3.1). This point was addressed and illustrated in simulated date in a recent paper by Ristic et al. [Rist 10b].

### 3.3.2 A new approach: the BET manager

#### Principle

The construction of the objective function based on the sole PENT seems to have two major flaws:

- Conceptually, the PENT is difficult to translate in an "operational" objective;
- Theoretically, its model suffers from several inconsistencies (see section 3.3.1).

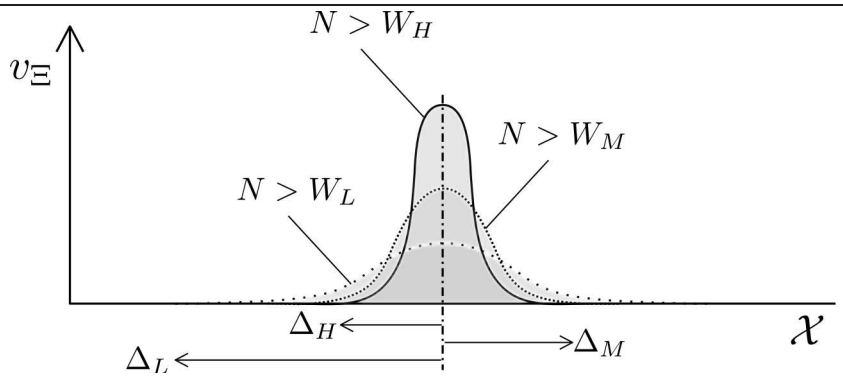
In consequence, the PENT manager performs poorly in certain situations as illustrated on simulated data in chapter 4.

The BET manager embodies a different approach than the PENT and aims at providing an efficient sensor management in situations where the sensor coverage is lacking, i.e. the combined FOVs cannot cover the whole state space simultaneously. The idea is to focus the sensors on a few points of the state space called *tracks*. The term "track" should not be interpreted in the classical sense since the PHD filter does not maintain such tracks, but rather as the presence of a target based on the *local* value of the PHD. The tracks are extracted in areas of the state space where the local weight exceeds a given threshold in a similar way as the weighted extractor does (see section 3.1.2). More precisely, three kind of tracks may be extracted in a PHD:

1. First, *high* tracks are extracted when at least $W_H$ worth of weight can be extracted in a region centered around a peak with a radius smaller than $\Delta_H$;
2. Then, *medium* tracks are extracted when at least $W_M$ worth of weight can be extracted in a region centered around a peak with a radius smaller than $\Delta_M$;
3. Finally, *low* tracks are extracted when at least $W_L$ worth of weight can be extracted in a region centered around a peak with a radius smaller than $\Delta_L$.

Since the local value of the PHD provides information on the target number and localization (if any), the tracks can be exploited just the same way. A low track indicates the existence of a target with low probability and with uncertain localization, both the probability of existence and the precision of the localization increasing with the track level. The values of the weight $W_H > W_M > W_L$ and distance $\Delta_H < \Delta_M < \Delta_L$ parameters are, of course, critical to the proper design of the sensor manager. This point will be discussed later in this section.

**Figure 3.14** High, medium and low tracks




All the tracks are not necessarily worth being focused on. Depending on the context of the surveillance activity and the geographical features of the ground, it is assumed that the state space can be decomposed in *exploration* and *tracking zones*, in which the main objective is respectively *target detection* and *target localization*. Thus, the *focus tracks* are defined as follows:

Table 3.1: Focus tracks

| Zone | Track level | | |
|---|---|---|---|
| | Low | Medium | High |
| Exploration | Yes | No | No |
| Tracking | Yes | Yes | No |

The qualitative analysis in section 2.3.1 suggests that, if a focus track is covered by at least one sensor:

- If there is a true target behind the track, the weight of the track will eventually increase and the local shape of the PHD will sharpen; thus, the level of the track will increase as well;
- Conversely, if the track is a false alarm, its weight will eventually decrease and the track will disappear.

Thus, a first objective of the surveillance can be stated as follows:

**Principle 1**: in exploration as well as tracking zones, the sensors should cover as many focus tracks as possible, until they either disappear or become non-focus tracks.

The difference between the exploration and tracking zones being, as indicated by

table 3.1, the threshold between "uncertain targets" (focus tracks) and "certain targets" (non-focus tracks). Besides, if a non-focus track is not covered by any sensors, the local shape of the PHD should "flatten" during the successive time update steps as the uncertainty about the localization of the target grows. That is, the level of the track should eventually decrease and the track should be granted the "focus" status again.

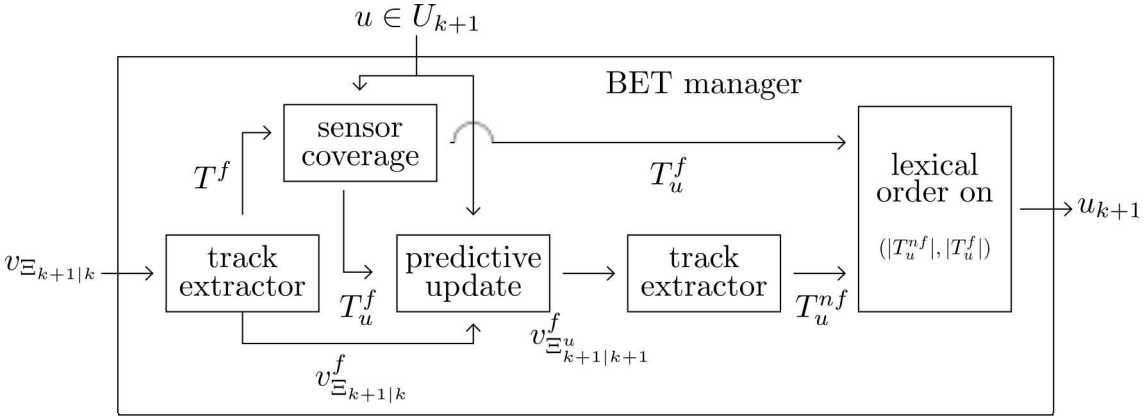
Yet this principle is not sufficient, because it does not priorize the actions of the sensors if the number of focus tracks exceeds the covering capacities of the sensors. In order to avoid the sensors from wasting their resources on too many tracks, it was decided that the sensors should focus on the "most promising" tracks, that is, those whose level should increase were they covered by at least on sensor. This leads to the second general objective:

**Principle 2**: the best controls are those which are likely to promote the highest number of focus tracks to the non-focus status.

Depending on the quality of the sensors, though, a focus track could require a sensor coverage during several successive iterations before being promoted to the non-focus status. Hence the last principle:

**Principle 3**: among the controls that are likely to promote the same number of focus tracks to the non-focus status, the best ones are those covering the highest number of focus tracks.

**Figure 3.15** Data flow of the BET manager




The data flow of the BET manager can be summarized as follows:

1. The non-focus tracks are extracted from the time udpated PHD $v_{\Xi_{k+1|k}}$, and

the corresponding weight is removed from the PHD. The resulting PHD $v^f_{\Xi_{k+1|k}}$ has thus been removed from "certain targets".

2. The focus tracks $T^f$ are extracted from $v^f_{\Xi_{k+1|k}}$, but the corresponding weight is kept in the PHD.

3. For each available control $u \in U_{k+1}$, the tracks among $T^f$ that are covered by the sensors under control $u$ are stored in $T^f_u$.

4. For each available control $u \in U_{k+1}$, the PIMS is constructed according to definition 3.5 with $T^f_u$ as input. Then, the predictive PHD is computed according to proposition 3.3, with the reduced $v^f_{\Xi_{k+1|k}}$ as input. The resulting PHD is $v^f_{\Xi^u_{k+1|k+1}}$.

5. For each available control $u \in U_{k+1}$, the non-focus tracks are extracted from $v^f_{\Xi^u_{k+1|k+1}}$ and stored in $T^{nf}_u$.

6. The control with the highest number of non-focus tracks $|T^{nf}_u|$ is selected. If there are ties, the control which covered the highest number of focus tracks $|T^f_u|$ is selected. If there are still ties, the control is chosen at random.

The BET manager share the same basic features as the PENT manager (see figures 3.9 and 3.15) by following the pattern "object extraction → predictive update → evaluation of the predictive PHD". The key difference lies in the nature of the extracted objects. While the PENT considers the estimated targets that can be extracted from the time updated PHD, the BET ignores the well-extracted targets and directs the sensors toward unknown regions by artificially creating tracks which can be seen as "weaker" versions of targets.

### Computational cost: qualitative analysis

In the PENT as well as in the BET manager, the computational cost lies mainly in the predictive update. Clearly, the computational cost of the exact predictive multi-sensor PHD (proposition 3.3) increases dramatically with the number of ideal measurements, thus with the number of extracted targets/tracks and the overlapping of the FOV configuration (recall that a target/track will produce one ideal measurement per sensor whose FOV covers its position). Thus, it is almost always "safer" to approximate the predictive update with the much lighter ICA (corollary 3.1).

In any case, the computational cost of the BET manager can be significantly reduced by discarding some controls $u \in U_{k+1}$ *before* the predictive update. Indeed,

recall from the second principle before and the data flow (figure 3.15) that the first selection criteria is the number of non-focus tracks in the predictive update $v^f_{\Xi^u_{k+1|k+1}}$. These non-focus tracks must necessarily stem from some increase in the value of the PHD during the predictive step, since by construction the input PHD $v^f_{\Xi_{k+1|k}}$ has been removed from its non-focus tracks. Consider an available control $u \in U_{k+1}$ that covers no focus tracks ($T^f_u = \emptyset$). Then, the ideal measurement set will be empty as well and proposition 3.2 with $Z^{WE}_{k+1} = \emptyset$ gives:

$$\begin{aligned} v^f_{\Xi^u_{k+1|k+1}}(x|Z_{1:k}) &= \beta[\delta_\emptyset, \delta_x]K_{\mathcal{X}}^{-1} \\ &= \prod_{j=1}^{S}(1 - p_u^{d,j}(x))v^f_{\Xi_{k+1|k}}(x|Z_{1:k}) \\ &\leqslant v^f_{\Xi_{k+1|k}}(x|Z_{1:k}) \end{aligned}$$

Thus, no non-focus tracks will be extracted from the predictive PHD $v^f_{\Xi^u_{k+1|k+1}}$, that is, $|T^{nf}_u| = 0$ with probability one. Consequently, the available controls that covers no focus tracks can be discarded without loss of performance for the BET manager. Arguably, the control that is likely to produce the highest number of non-focus tracks is among the controls that covers a large number of focus tracks. Thus, the computational cost of the BET manager could be further reduced by processing the available controls by decreasing number of covered focus tracks, and stop whenever the computing time or the number of processed controls exceeds a given limit. However, there is no guarantee that the control producing the highest number of non-focus tracks will eventually be processed and therefore be selected.

### Design challenges

The BET manager is a first approach in sensor management and was mainly designed upon the author's understanding of the mechanics and flaws of the PENT manager. Arguably, the conception of the BET manager brings two major issues:

- the predictive update equation depends on some extraction process that falls outside the PHD framework;
- the values of weight $W_H$, $W_M$ $W_L$ and distance $\Delta_H$, $\Delta_M$, $\Delta_L$ parameters are critical to the BET peformance.

The first issue is shared with the PENT manager and does actually exceed the sensor management framework. As discussed previously, the predictive PHD is built as an expectation (definition 3.3) that requires an extreme simplification to become tractable (see section 3.2 for a detailed explanation). An interesting lead for further

work could be the design of a predictive PHD in an entirely different manner, but this point is beyond the scope in this thesis.

The second issue, however, is specific to the BET manager. The problem lies in the fact that the track thresholds depend on many parameters of the system, notably the target motion model and the sensor parameters. Consider, for example, the task of setting the parameters $(W_H, \Delta_H)$. Suppose that a high track lies in a tracking zone with no geographical elements. Since the track is high, it is a non-focus track; thus it will be ignored by the sensors. The track motion is relatively free of constraints, thus the local shape of the PHD should flatten significantly during the successive iterations when the track is not covered. Thus, one may think of setting restrictive values for the high level (say, $W_H = 0.9$, $\Delta_H = 10$) to ensure that an unobserved high track is quickly demoted to the medium level and therefore becomes a focus track again. Consider now a medium track moving along a road in a tracking zone. With the values above, the sensors will waste resources trying to raise the level of this focus track, even if, the motion of the track being relatively constrained, excessive use of sensors in this region may be superfluous. This particular situation would require a lower thereshold for high level tracks (say, $W_H = 0.8$, $\Delta_H = 30$). Not surprisingly, the parametrization of the threshold for the medium track is even more critical, since it influences the sensor management in both exploration and tracking zones. A solution may be to make the thresholds dependent on the track position in the surveillance region (or, more generally, on the track state), but this has not been explored yet.

## 3.4 Conclusion

This chapter covered all the elements pertaining to the design of a simple sensor manager within the PHD framework. First of all, the target extraction process was discussed and a solution based on the extraction of weight in the PHD was implemented. Then followed the rigorous construction of the predictive PHD, based on the PIMS proposed by Mahler. Similarly to the data update step in chapter 2, the predictive step was simplified without approximation thanks to a joint partitioning method. The last part was devoted to a discussion about Mahler's PENT manager and its inconsistencies on simple examples, followed by the proposition of a new sensor manager - the BET manager - based on a different and more "operational" approach to the sensor management problem.

CHAPTER 4

# Implementation and results

This chapter deals with the practical implementation of a multi-sensor PHD filter (chapter 2) and a BET manager (chapter 3) in a detection and tracking problem. The algorithms were implemented in Matlab for the most part, some routines were written in embedded C code. The first section deals with the modelization of a surveillance scenario. The next section describes the SMC implementation of the PHD filter and the BET manager. Finally, the last section provides the main simulation results.

## 4.1 Scenario modelization

In this chapter, the sensors are assumed to be fast enough compared to the targets so that the time step in the filtering flow (see figure 1) is driven by the target motion model. The duration between two time steps is arbitrarily set at $\Delta_t = 1\ s$.

### 4.1.1 Target modelization

#### State space

A target state $x \in \mathcal{X}$ has two position variables and two velocity variables. The state space $\mathcal{X} \subset \mathbb{R}^4$ is the bounded subset such that $x = [x^c, y^c, \dot{x}^c, \dot{y}^c]^T \in \mathcal{X}$ if and only if:

$$\begin{cases} (x^c, y^c) \in \mathcal{R} \stackrel{def}{=} [x^c_{min}\ x^c_{max}] \times [y^c_{min}\ y^c_{max}] \\ \sqrt{(\dot{x}^c)^2 + (\dot{y}^c)^2} \leqslant v_{max} \end{cases} \tag{4.1}$$

where $x^c_{min}, x^c_{max}, y^c_{min}, y^c_{max}, v_{max}$ are given boundaries. Their values should be adapted to the underlying physical problem. In this thesis, $(x^c, y^c)$ are coordi-

nates in meters and $\mathcal{R}$ is arbitrarily set as an area of 1 square kilometer, i.e. $\mathcal{R} = [0\ 1000] \times [0\ 1000]\ m^2$). Besides, since the targets are ground-based and $\Delta_t = 1\ s$, one may safely assume that a target will not move more than 10 meters between successive time steps, thus the maximum velocity is set at $v_{max} = 10\ m.s^{-1}$.

Note that a true target has an implicit fifth state variable, namely its label. A label is uniquely attributed to each target when it is created for the whole duration of the simulation - i.e. a label is unavailable if it has already been granted to a previous target, even if this target has already died. The true targets at time step $x$ are gathered in the set $X_k = \{x_k^i\}_{i \in I(k)}$ where $I(k) \subset \mathbb{N}^*$ is the set of current true target indices.

### Free model

This first target model is "generic" in the sense that the target motion is not influenced by the local topography of the surveillance region. This simple model was specifically designed for comparison purpose between filtering techniques and is based on the following assumptions:

- the number of newborn targets at each time step is Poisson;
- the newborn targets are uniformly distributed in the state space;
- the target motion model is the *near-constant velocity* (NCV) model [Li R 03];
- a living targets die when (and only when) it leaves the surveillance region.

The pseudo-code of the free model is given in algorithm 1.

Note that the parameters of the model, the birth intensity $\lambda_{k-1,k}^b$ and the standard deviations $\sigma_k^x$, $\sigma_k^y$, are independent of the target state but may depend on the time step. It is interesting to have a large variation in the target number during the simulation, one way to achieve this is to set a periodic birth intensity as follows:

$$\lambda_{k-1,k}^b = \lambda \cos\left(2\pi \frac{k}{T_\lambda}\right) \tag{4.2}$$

Typical values of the parameters are $\lambda = \frac{1}{15}$, $T_\lambda = 80$.

**Algorithm 1** Free target model (time $k$)

**input:** Target set from previous iteration: $\{x_i\}_{i\in I(k-1)}$
**output:** Target set from current iteration: $\{x_i\}_{i\in I(k)}$

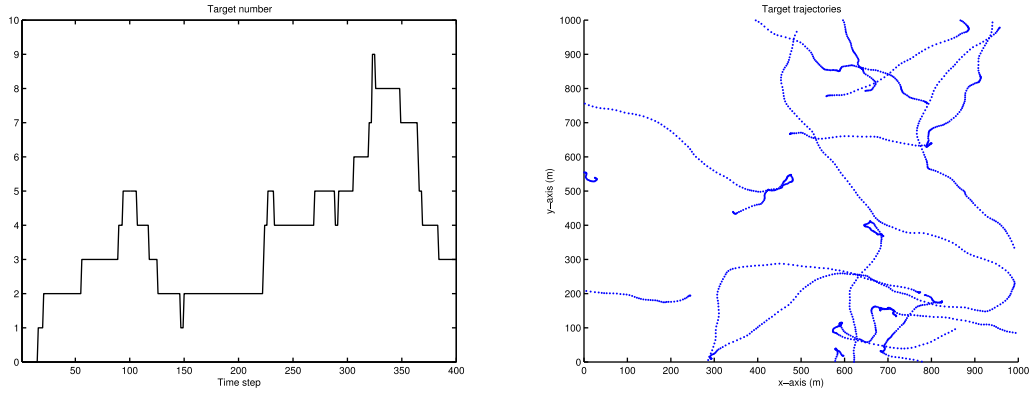
Target evolution
**for** $i \in I(k-1)$ **do**
  Target motion
  Compute white noise acceleration: $a \sim \mathcal{N}\left(\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}\sigma_k^x\\ \sigma_k^y\end{bmatrix}^2\right)$
  Compute new target state:
  $$x_{i,k} \leftarrow \begin{bmatrix}1 & 0 & \Delta_t & 0\\ 0 & 1 & 0 & \Delta_t\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix} x_{i,k-1} + \begin{bmatrix}(\Delta_t)^2/2 & 0\\ 0 & (\Delta_t)^2/2\\ \Delta_t & 0\\ 0 & \Delta_t\end{bmatrix} a$$
  Velocity normalization
  **if** $\sqrt{(\dot{x}_{i,k}^c)^2 + (\dot{y}_{i,k}^c)^2} > v_{max}$ **then**
    $\dot{x}_{i,k}^c \leftarrow \dot{x}_{i,k}^c \frac{v_{max}}{\sqrt{(\dot{x}_{i,k}^c)^2+(\dot{y}_{i,k}^c)^2}}$
    $\dot{y}_{i,k}^c \leftarrow \dot{y}_{i,k}^c \frac{v_{max}}{\sqrt{(\dot{x}_{i,k}^c)^2+(\dot{y}_{i,k}^c)^2}}$
  **end if**

  Target survival
  **if** $x_{i,k}^c \notin [x_{min}^c\ x_{max}^c]$ **and** $y_{i,k}^c \notin [y_{min}^c\ y_{max}^c]$ **then**
    Discard target $i$: $I(k-1) \leftarrow I(k-1) \setminus i$
  **end if**
**end for**

Target birth
Compute newborn target number: $N_b \sim Poisson(\lambda_{k-1,k}^b)$
Select next $N_b$ available labels: $I_b(k)$
**for** $i \in I_b(k)$ **do**
  Compute new position coordinates:
  $x_{i,k}^c \sim \mathcal{U}([x_{min}^c\ x_{max}^c])$, $y_{i,k}^c \sim \mathcal{U}([y_{min}^c\ y_{max}^c])$
  Compute new velocity coordinates:
  $\theta \sim \mathcal{U}([-\pi\ \pi])$, $v \sim \mathcal{U}([0\ v_{max}])$
  $\dot{x}_{i,k}^c \leftarrow v\cos(\theta)$, $\dot{y}_{i,k}^c \leftarrow v\sin(\theta)$
**end for**
Update set of living target labels: $I(k) \leftarrow I(k-1) \cup I_b(k)$

For simplicity's sake, the standard deviations $\sigma_k^x$ and $\sigma_k^y$ are fixed, the typical values are $\sigma_k^x = \sigma_k^y = 1$. A free scenario of 400 time steps looks as follows:

**Figure 4.1** Example of free scenario




### Ground-based model

This second target model was designed for the specific purpose of sensor management evaluation on more "realistic" scenarii than those based on the free model above. The structure of the ground-based model is similar to the free model, but the birth and evolution of the targets are influenced by geographical elements in the surveillance region.

The influence of nearby elements on the target evolution is encapsulated in the *influence vector* and *influence parameter*:

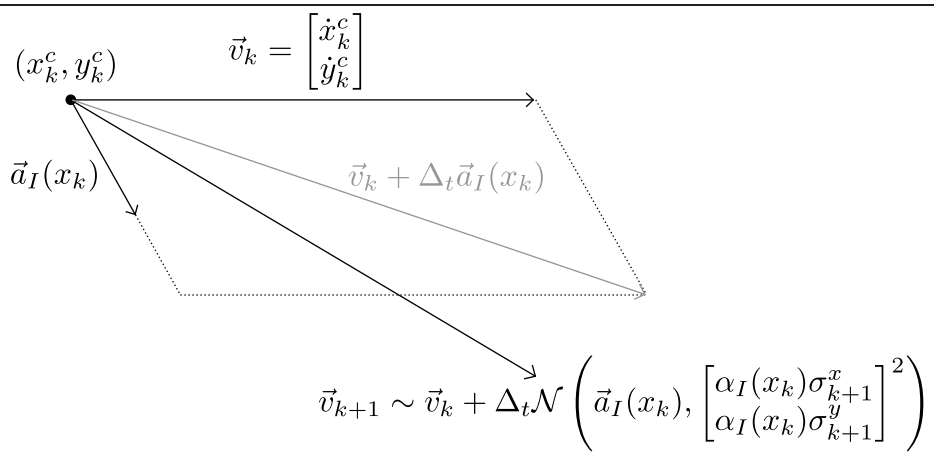
$$\begin{aligned}(a_I, \alpha_I) \ : \ \mathcal{X} &\to \mathbb{R}^2 \times [0\ 1] \\ x &\mapsto (a_I(x), \alpha_I(x))\end{aligned} \tag{4.3}$$

The influence vector $a_I(x)$ has the same unit as a acceleration and indicates the "favored motion" of a target in state $x$ given the nearby elements such as:

- roads: targets getting closer to a road tend to follow the road as a general direction;
- obstacles: targets cannot reach these "forbidden zones".

A target $x_k$ evolves according to the NCV model in algorithm 1 except that the acceleration is drawn with mean $a_I(x_k)$ and with reduced variance $[\alpha_I(x)\sigma_{k+1}^x\ \alpha_I(x)\sigma_{k+1}^y]^T$:

**Figure 4.2** Groud-based influence on target $x_k$




The influence parameter is tuned such that it decreases when the target becomes close to geographical elements, that is, a target is less likely to "wander around" in the vicinity of roads or obstacles.

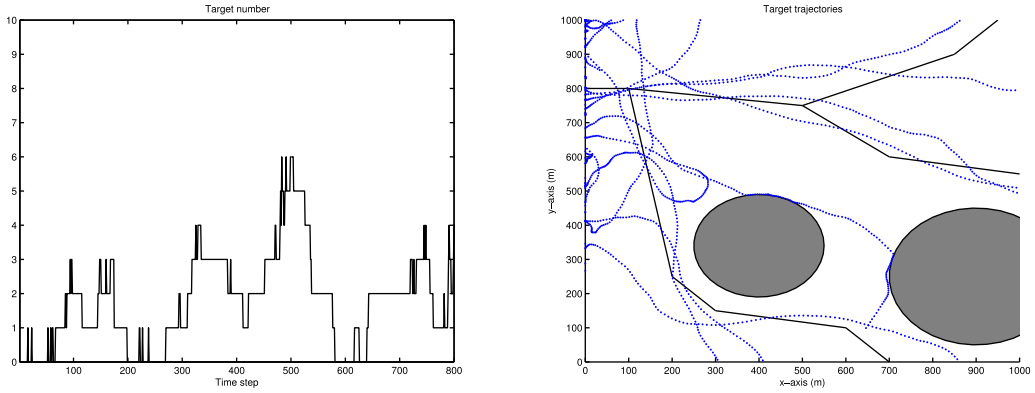
Besides, after a target has moved or has been created, one must check that the target is not inside an obstacle. In this case, its position is moved to the nearest point outside the obstacle, and its velocity vector is modified so that the target moves along the obstacle rather than bump into it in the next iterations.

The number of newborn targets is still Poisson, but the targets are not necessarily drawn uniformly in the state space. For example, the newborn targets may be created along the edges of the surveillance region, with a velocity vector pointing inward. In the following scenario (figure 4.3) the newborn targets are distributed according to:

$$\begin{cases} x_{k+1}^c = 0 & \\ y_{k+1}^c \sim \mathcal{N}(800, 100) & (y_{min}^c \leqslant y_{k+1}^c \leqslant y_{max}^c) \\ \dot{x}_{i,k+1}^c = v\cos(\theta) & \\ \dot{y}_{i,k+1}^c = v\sin(\theta) & (\theta \sim \mathcal{U}\left(\left[-\frac{\pi}{2}\ \frac{\pi}{2}\right]\right),\ v \sim \mathcal{U}\left([0\ v_{max}]\right)) \end{cases}$$

that is, the targets are coming from the left and presumably from the road. The other parameters $\lambda$, $T_\lambda$, $\sigma_{k+1}^x$ and $\sigma_{k+1}^y$ have the same values as in the free scenario presented before.

**Figure 4.3** Example of ground-based scenario (roads in black, obstacles in gray)

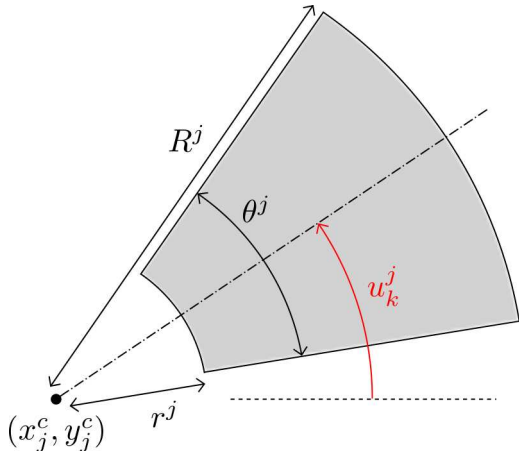



The main interest of this model is its ability to simulate a broad range of ground-based target behaviors with a limited computational cost. Compared to the free model, the ground-based model requires the additional computation of the influence vector for each target. This step, however, can have a significant cost in regions with several elements (e.g. road crossings). The field of influence vectors being static, it can be approximated by an offline evaluation of the field on a grid-based discretization of the state space. A relatively coarse grid is sufficient for a proper implementation of the model (respectively 200 and 10 knots in the position and velocity dimensions).

### 4.1.2 Sensor modelization

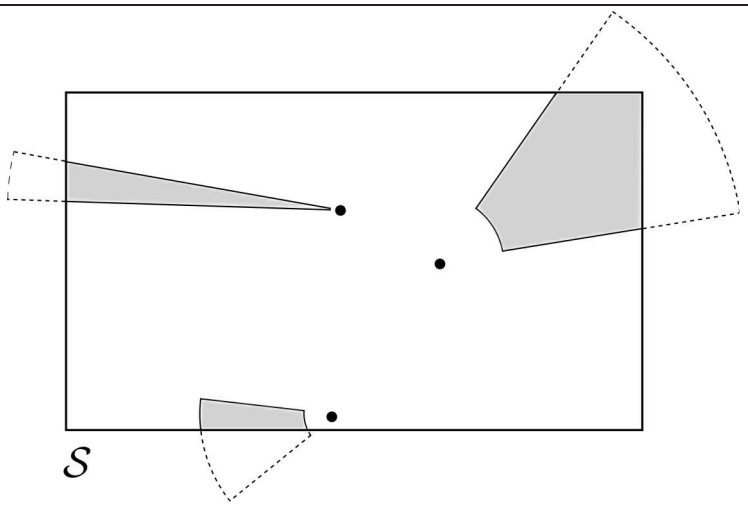
In this thesis, the sensors are ground-based and their position in the surveillance region is fixed. The FOV $F_u^j \subseteq \mathcal{X}$ is determined by the direction of focus $u_k^j$:

**Figure 4.4** Shape of sensor FOV $F_u^j$ in the surveillance region (gray area)



where $\theta^j$, $r^j$ and $R^j$ are constant parameters of sensor $j$. Note that the FOV shape may be significantly truncated in examples such as the following:

**Figure 4.5** Example of truncated FOVs



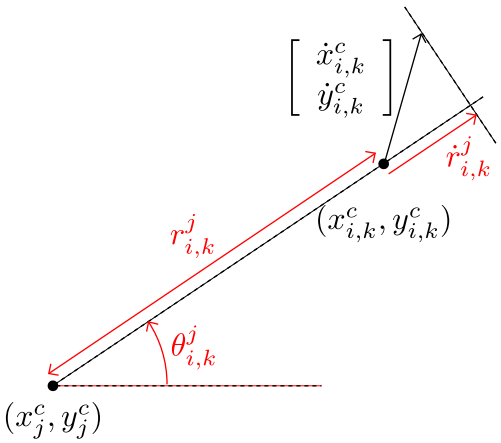

It is important to note that the FOVs are defined in the position subspace only, i.e. the detection probability for a target $x$ does not depend of its velocity $(\dot{x}^c, \dot{y}^c)$. The detection probability is considered uniform inside the FOV and, for any available direction $u_k^j$:

$$p_u^{d,j}(.) = p^{d,j} 1_{F_u^j}(.) \tag{4.4}$$

where $p^{d,j}$ is a constant parameter of sensor $j$, typically around 0.9.

The dimension of the observation space $\mathcal{Z}^j$ depends on the *class* of the sensor, the ideal measurement (see definition 3.4) is built as follows:

**Figure 4.6** Ideal measurement $\rho_k^j(x_{i,k})$




If sensor $j$ is $1^{st}$ class:

$$\rho_k^j(x_{i,k}) = [r_{i,k}^j, \theta_{i,k}^j]^T$$

If sensor $j$ is $2^{nd}$ class:

$$\rho_k^j(x_{i,k}) = [r_{i,k}^j, \theta_{i,k}^j, \dot{r}_{i,k}^j]^T$$

Clearly, the measurement functions are *not* bijective. Recall from chapter 3 that the injectivity is needed for the theoretical construction of ideal measurement sets (equation (3.8)), while the bijectivity is needed for the construction of the predictive observation RFSs (equation (3.9)). However, these issues are hardly relevant in practical implementations. First, an ideal measurement set would be ill-defined if two different targets $x_1$, $x_2$ shared the same ideal measurement ($\rho_k^j(x_1) = \rho_k^j(x_2)$) this would imply (figure 4.6) that they shared the same position coordinates in the surveillance region ($x_1^c = x_2^c$ and $y_1^c = y_2^c$) which is unlikely since there is no spawning and the targets evolve independently from each other. Then, the construction of predictive RFSs requires only that, for any ideal measurement $z$, the quantity $p_u^{d,j}((\rho_{k+1}^j)^{-1}(z))$ is properly defined. On the one hand the measurement functions restricted to the position coordinates in $\mathcal{X}$ and corestricted to the polar coordinates in $\mathcal{Z}^j$ are clearly bijective, on the other hand the detection probability $p_u^{d,j}(x)$ does not depend on the velocity coordinates $(\dot{x}^c, \dot{y}^c)$, thus $p_u^{d,j}((\rho_{k+1}^j)^{-1}(z))$ is well-defined.

The noise in the measurement process is assumed to be additive white Gaussian noise on each dimension. If measurement $z_{m,k}^j = [\hat{r}_{m,k}^j, \hat{\theta}_{m,k}^j, \hat{\dot{r}}_{m,k}^j]^T$ stems from $x_{i,k}$ then:

$$\begin{cases} \hat{r}_{m,k}^j \sim \mathcal{N}(r_{i,k}^j, (\sigma_r^j)^2) \\ \hat{\theta}_{m,k}^j \sim \mathcal{N}(\theta_{i,k}^j, (\sigma_\theta^j)^2) \\ \hat{\dot{r}}_{m,k}^j \sim \mathcal{N}(\dot{r}_{i,k}^j, (\sigma_{\dot{r}}^j)^2) \end{cases} \tag{4.5}$$

where parameters $\sigma_r^j$, $\sigma_\theta^j$ and $\sigma_{\dot{r}}^j$ (if $2^{nd}$ class sensor) are assumed constant. Note that there is no correspondance between measurement subscripts $m$ and target subscripts $i$ since the mapping between measurements and true targets is unknown.

The special case where the drawn measurement falls "outside the FOV" (i.e. $z_k^j \notin \rho_k^j(F_k^j)$) must be addressed in order to avoid inconsistencies in the measurement process. Two leads were followed:

- resample the measurement;
- "move" the measurement to the closest point in $\rho_k^j(F_k^j)$.

The first method is very simple since it does not require any additional algorithm, yet it may never converge if the FOV is very small (typically when the whole FOV is heavily truncated, see figure 4.5). The second method brings a stronger bias in the distribution of the sampled measurement, especially when the origin target is close to the FOV's edges. However, it is computationally safer and therefore was selected.

The likelihood function $L_k^{\cdot,j}(.)$ is given by:

$$\begin{cases} L_k^{z^j_{m,k},j}(x_{i,k}) = \dfrac{e^{-\frac{1}{2}\left(\left(\frac{r^j_{i,k}-\hat{r}^j_{m,k}}{\sigma^j_r}\right)^2+\left(\frac{\theta^j_{i,k}-\hat{\theta}^j_{m,k}}{\sigma^j_\theta}\right)^2\right)}}{2\pi\sigma^j_r\sigma^j_\theta} & (1^{st}\ class) \\ L_k^{z^j_{m,k},j}(x_{i,k}) = \dfrac{e^{-\frac{1}{2}\left(\left(\frac{r^j_{i,k}-\hat{r}^j_{m,k}}{\sigma^j_r}\right)^2+\left(\frac{\theta^j_{i,k}-\hat{\theta}^j_{m,k}}{\sigma^j_\theta}\right)^2+\left(\frac{\dot{r}^j_{i,k}-\hat{\dot{r}}^j_{m,k}}{\sigma^j_{\dot{r}}}\right)^2\right)}}{(2\pi)^{\frac{3}{2}}\sigma^j_r\sigma^j_\theta\sigma^j_{\dot{r}}} & (2^{nd}\ class) \end{cases} \tag{4.6}$$

The false alarm process is a classical model for radar sensors. The reduction of the FOV to the position subspace (the gray area in figure 4.4) is partitioned in elementary cells (one degree of arc wide and one meter long) such that the occurence of a false alarm in each cell is an independent Bernouilli process with the *probability of false alarm* $p_k^{fa,j}$ as parameter. Since the number of cells is very large and the probability of false alarm very low (usually around $10^{-5}$), the number of false alarms is accurately approximated as Poisson with intensity:

$$\lambda_k^{c,j} = p_k^{fa,j}\frac{180.\theta^j}{\pi}(R^j - r^j) \tag{4.7}$$

and each false alarm is drawn uniformly in $\rho_k^j(F_k^j)$. The easiest way to deal with the truncated FOVs is to ignore them in the drawing process (i.e. draw the false alarm number according to (4.7) and distribute the false alarm measurements uniformly in the FOV shape), then discard the false alarms that falls outside the surveillance region. Note that the false alarm term in the expression of the cross-terms (see proposition 2.5) is independent of the measurement itself thanks to the uniform draw:

$$\forall z \in \rho_k^j(F_k^j),\ \lambda_k^{c,j}c_k^j(z)K_{\mathcal{Z}^j} = p_k^{fa,j} \tag{4.8}$$

**Algorithm 2** Measurement process from $2^{nd}$ class sensor $j$ (time $k$)

**input:** Target set from current iteration: $\{x_i\}_{i \in I(k)}$
**output:** Measurement set from sensor $j$: $Z_k^j = \{z_m^j\}_{m \in [1 \ m_k^j]}$

True measurements
Initialize measurement counter: $m \leftarrow 0$
**for** $i \in I(k)$ **do**
  Draw random variable: $p \sim \mathcal{U}([0\ 1])$
  **if** $x_{i,k} \in F_u^j$ **and** $p \leqslant p^{d,j}$ **then**
    Computation of a true measurement
    Increment measurement counter: $m \leftarrow m + 1$
    Compute ideal measurement: $\rho_k^j(x_{i,k}) \leftarrow [r_{i,k}^j, \theta_{i,k}^j, \dot{r}_{i,k}^j]^T$
    Compute noisy polar coordinates:
    $\hat{r} \sim \mathcal{N}(r_{i,k}^j, (\sigma_r^j)^2)$, $\hat{\theta} \sim \mathcal{N}(\theta_{i,k}^j, (\sigma_\theta^j)^2)$
    Compute noisy radial velocity: $\hat{\dot{r}} \sim \mathcal{N}(\dot{r}_{i,k}^j, (\sigma_{\dot{r}}^j)^2)$
    Store true measurement: $z_{m,k}^j \leftarrow [\hat{r}, \hat{\theta}, \hat{\dot{r}}]^T$

    Validity check of the true measurement
    **if** $z_{m,k}^j \notin \rho_k^j(F_u^j)$ **then**
      Move to closest valid measurement: $z_{m,k}^j \leftarrow \arg\min_{z \in \rho_k^j(F_u^j)} d_{\mathcal{Z}^j}(z, z_{m,k}^j)$
    **end if**
  **end if**
  Update measurement number: $m_k^j \leftarrow m$
**end for**

False alarms
Compute false alarm parameter: $\lambda_k^{c,j} \leftarrow p_k^{fa,j} \frac{180.\theta^j}{\pi}(R^j - r^j)$
Compute false alarm number: $m^{fa} \sim Poisson(\lambda_k^{c,j})$
**for** $m = 1$ **to** $m^{fa}$ **do**
  Computation of a false alarm measurement
  Compute random polar coordinates:
  $\hat{r} \sim \mathcal{U}([r^j\ R^j])$, $\hat{\theta} \sim \mathcal{U}([u_k^j - \frac{\theta^j}{2}\ u_k^j + \frac{\theta^j}{2}])$
  Compute random radial velocity: $\hat{\dot{r}} \sim \mathcal{U}([-v_{max}\ v_{max}])$
  Store temporary measurement: $z \leftarrow [\hat{r}, \hat{\theta}, \hat{\dot{r}}]^T$

  Validity check of the false alarm measurement
  **if** $z \in \rho_k^j(F_u^j)$ **then**
    Update measurement number: $m_k^j \leftarrow m_k^j + 1$
    Store false alarm measurement: $z_{m_k^j,k}^j \leftarrow z$
  **end if**
**end for**

## 4.2 Implementation of the multi-sensor PHD filter

### 4.2.1 GM vs. SMC methods

The equations of the PHD filter (2.14), (2.53), (2.63) and the predictive PHD (3.17), (3.23) have no closed-form expression in the general case, notably because of the presence of integrals over the state space $\mathcal{X}$. The integral appears explicitely in (2.14), in other equations through the cross-terms that are not derivated in a state point (e.g. $\beta[\delta_{\emptyset}, 1]$ or $\beta[\delta_{\{z^1,z^2\}}, 1]$). Since the beginning of the PHD filter, two different implementation techniques have been enjoying a wide popularity: the *Gaussian mixture PHD* (GMPHD) and the *sequential Monte Carlo PHD* (SMCPHD) .

The GMPHD filter is a closed form expression of the PHD equations in the particular case where the target dynamics and measurement model are linear Gaussian, although it can be adapted to a broader range of situations by replacing the Kalman filter equations in the GMPHD by their linearization as in the extended Kalmal filter, or their approximation as in the unscented Kalman filter [Pace 11]. This model assumes that the intensities of the birth and spawning RFS are Gaussian mixtures, and that the probabilities of target detection and target survival do not depend on the target state. Then, it can be shown that the time and data updated PHDs are also Gaussian mixtures with a closed form expression. The GMPHD has been used in numerous recent tracking algorithms [Pant 09, Lee 10, Lund 11] and an implementation of the GMCPHD, the extension of the GMPHD to the CPHD filter, is given in [Ulmk 10].

The SMCPHD filter has been first implemented by Vo et al. in [Vo 03, Vo 05]. As its name suggests, this method aims to apply SMC methods for Bayesian filtering [Douc 00] to PHDs rather than usual probability densities. Since the PHD is a first-order moment density, it is unnormalized and do not follow the usual Bayes recursion; Johansen et al. [Joha 05] studied and proved the convergence of the SMC implementation of the PHD under reasonable assumptions. The SMC implementation is specifically designed for highly nonlinear systems, but requires fine-tuning to be efficient. In the SMCPHD framework, the choice of importance functions for the prediction and the birth of particles is known to be a difficult task and an active research topic [Rist 10a]. Besides, the accuracy and the computational cost of the SMC implementation both increasing with the number of particles, its tuning is also critical to the quality of the SMC implementation.

Pace [Pace 11] compared the GMPHD and SMCPHD filters on scenarii with a constant number of targets evolving according to an interacting multiple model composed by a constant velocity model and a constant-turn model perturbed with random accelerations. The results show that the GMPHD filter outperforms the

SMCPHD filter in both estimate quality and computational cost. To be sure, the quality of the SMCPHD filter increases with the number of particles, but the computational cost also increases (linearly) and is not adapted to a large number of targets. In the scope of this thesis, however, the SMC implementation seemed more natural since the ground-based evolution model (see section 4.1.1) is highly nonlinear and the independence of the survival and detection probabilities with the target state seems incompatible with the FOV-oriented sensor management. The SMC implementation presented in the next section is largely based on the SMCPHD filter given in [Vo 05].

### 4.2.2 SMC implementation

The principle of the SMCPHD is to propagate a set of weighted particles approximating the successive predicted $v_{\Xi_{k+1|k}}$ and posterior $v_{\Xi_{k+1|k+1}}$ PHDs rather than the full PHDs. At each time $k$:

$$v_{\Xi_{k+1|k}}(.|Z_{1:k}) \simeq \sum_{l=1}^{L_{k+1|k}} w_{k+1|k}^{(l)} \delta_{x_{k+1|k}^{(l)}}(.) \tag{4.9}$$

$$v_{\Xi_{k+1|k+1}}(.|Z_{1:k+1}) \simeq \sum_{l=1}^{L_{k+1}} w_{k+1}^{(l)} \delta_{x_{k+1}^{(l)}}(.) \tag{4.10}$$

This section describes the successive operations of the SMCPHD at time $k+1$, whose goal is to modify the set of weighted particles as follows:

$$\{x_k^{(l)}, w_k^{(l)}\}_{l\in[1\ L_k]} \longrightarrow \{x_{k+1}^{(l)}, w_{k+1}^{(l)}\}_{l\in[1\ L_{k+1}]} \tag{4.11}$$

The different operations can be summarized as follows:

1. Evolution: $\{x_k^{(l)}, w_k^{(l)}\}_{l\in[1\ L_k]} \rightarrow \{x_{k+1|k}^{(l)}, w_{k+1|k}^{(l)}\}_{l\in[1\ L'_k]}$;
2. Model-based birth: $\{x_{k+1|k}^{(l)}, w_{k+1|k}^{(l)}\}_{l\in[1\ L'_k]} \rightarrow \{x_{k+1|k}^{(l)}, w_{k+1|k}^{(l)}\}_{l\in[1\ L'_k+J_{k+1}]}$;
3. Selection of control $u_{k+1}^{opt} \in U_{k+1}$ and production of measurement set $Z_{k+1}$;
4. Measurement-based birth: $\{x_{k+1|k}^{(l)}, w_{k+1|k}^{(l)}\}_{l\in[1\ L'_k+J_{k+1}]} \rightarrow \{x_{k+1|k}^{(l)}, w_{k+1|k}^{(l)}\}_{l\in[1\ L_{k+1|k}]}$;
5. Weight update: $\{x_{k+1|k}^{(l)}, w_{k+1|k}^{(l)}\}_{l\in[1\ L_{k+1|k}]} \rightarrow \{x_{k+1|k}^{(l)}, \tilde{w}_{k+1|k}^{(l)}\}_{l\in[1\ L_{k+1|k}]}$;
6. Resampling: $\{x_{k+1|k}^{(l)}, \tilde{w}_{k+1|k}^{(l)}\}_{l\in[1\ L_{k+1|k}]} \rightarrow \{x_{k+1}^{(l)}, w_{k+1}^{(l)}\}_{l\in[1\ L_{k+1}]}$;

Note the presence of two particle birth steps. The first step follows the target creation model and adds the appropriate weight in the filter. It should be sufficient in theory, but the filter was impractical as too few newborn particles appeared in the vicinity ot the measurements. Thus, additional particles are created around the measurements to ensure that at least a few particles remain around after the resampling. This solution is somewhat unsatisfying and should be corrected in future improvements of the filter.

For simplicity's sake, the operations are described assuming that the targets follow the free motion model. Should the opposite occur, one must modify the prediction and birth steps to account for the influence parameters and the obstacles as explained in section 4.1.1.

### Particle evolution

This operation aims to approximate the evolution process described by the integral part in the time update equation (2.14), that is, the evolution of surviving targets from the previous iteration. Assuming that there is no spawning, the new particles should be drawn as follows [Vo 05]:

$$\forall l \in [1\ L_{k-1}], x_{k+1|k}^{(l)} \sim q_{k+1}(.|x_k^{(l)}, Z_{k+1}) \tag{4.12}$$

$$\forall l \in [1\ L_{k-1}], w_{k+1|k}^{(l)} = \frac{p_{k,k+1}^{s}(x_k^{(l)}) f_{k,k+1}^{t}(x_{k+1|k}^{(l)}|x_k^{(l)})}{q_{k+1}(x_{k+1|k}^{(l)}|x_k^{(l)}, Z_{k+1})} w_k^{(l)} \tag{4.13}$$

where $q_{k+1}(.|x_k^{(l)}, Z_{k+1})$ is an appropriate importance function (see appendix B for a general description of importance sampling). The target motion model (especially the ground-based one) proved to be too challenging to design a proper importance function; besides, the current measurement set $Z_{k+1}$ is not available since the sensor manager has not chosen the multi-control yet. Thus the particles are drawn according to the target motion model itself.

**Algorithm 3** Particle evolution (time $k+1$)

**input:** Particle set from previous iteration: $\{x_k^{(l)}, w_k^{(l)}\}_{l\in[1\ L_k]}$
**output:** Evolved particle set from current iteration: $\{x_{k+1|k}^{(l)}, w_{k+1|k}^{(l)}\}_{l\in[1\ L'_k]}$

Particle evolution
**for** $l = 1$ **to** $L_k$ **do**
  Particle motion
  Compute white noise acceleration
  $a \sim \mathcal{N}\left(\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}\sigma_{k+1}^x\\ \sigma_{k+1}^y\end{bmatrix}^2\right)$
  Compute new particle state:
  $$x_{k+1|k}^{(l)} \leftarrow \begin{bmatrix}1 & 0 & \Delta_t & 0\\ 0 & 1 & 0 & \Delta_t\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix} x_k^{(l)} + \begin{bmatrix}(\Delta_t)^2/2 & 0\\ 0 & (\Delta_t)^2/2\\ \Delta_t & 0\\ 0 & \Delta_t\end{bmatrix} a$$
  Velocity normalization
  **if** $\sqrt{(\dot{x}_{k+1|k}^{(l),c})^2 + (\dot{y}_{k+1|k}^{(l),c})^2} > v_{max}$ **then**
    $\dot{x}_{k+1|k}^{(l),c} \leftarrow \dot{x}_{k+1|k}^{(l),c} \frac{v_{max}}{\sqrt{(\dot{x}_{k+1|k}^{(l),c})^2+(\dot{y}_{k+1|k}^{(l),c})^2}}$
    $\dot{y}_{k+1|k}^{(l),c} \leftarrow \dot{y}_{k+1|k}^{(l),c} \frac{v_{max}}{\sqrt{(\dot{x}_{i,k}^{c})^2+(\dot{y}_{k+1|k}^{(l),c})^2}}$
  **end if**

  Particle survival
  **if** $x_{k+1|k}^{(l),c} \notin [x_{min}^c\ x_{max}^c]$ **and** $y_{k+1|k}^{(l),c} \notin [y_{min}^c\ y_{max}^c]$ **then**
    Discard particle $l$
  **end if**
**end for**
Reorder remaining particle labels: $[1\ L_k] \rightarrow [1\ L'_k]$

### Model-based birth

This operation aims to approximate the birth process described by the non-integral part in the time update equation (2.14). The newborn particles should be drawn as follows [Vo 05]:

$$\forall l \in [L'_k + 1\ L'_k + J_{k+1}], x_{k+1|k}^{(l)} \sim p_{k+1}(.|Z_{k+1}) \tag{4.14}$$

$$\forall l \in [L'_k + 1\ L'_k + J_{k+1}], w_{k+1|k}^{(l)} = \frac{1}{J_{k+1}} \frac{\lambda_{k,k+1}^b b_{k,k+1}(x_{k+1|k}^{(l)})}{p_{k+1}(x_{k+1|k}^{(l)}|Z_{k+1})} \tag{4.15}$$

where $p_{k+1}(.|Z_{k+1})$ is an appropriate importance function. The current measurement set $Z_{k+1}$ is not available since the sensor manager has not chosen the multi-control yet. Thus, the particles are drawn according to the normalized birth intensity $b_{k,k+1}(.)$. A sound choice in the particle number is to keep the ratio of particles per target as constant as possible [Vo 05], thus the number of newborn particles is set as proportional to the expected number of newborn targets:

$$J_{k+1} = round(\rho_T \lambda^b_{k,k+1}) \tag{4.16}$$

where $\rho_T$ is the desired particle-per-target ratio. Note that the total weight brought by the newborn particles is the expected number of newborn targets:

$$\sum_{l=L'_k+1}^{L'_k+J_{k+1}} \frac{1}{J_{k+1}} \frac{\lambda^b_{k,k+1} b_{k,k+1}(x^{(l)}_{k+1|k})}{b_{k,k+1}(x^{(l)}_{k+1|k})} = \sum_{l=L'_k+1}^{L'_k+J_{k+1}} \frac{\lambda^b_{k,k+1}}{J_{k+1}} = \lambda^b_{k,k+1} \tag{4.17}$$

**Algorithm 4** Model-based particle birth (time $k+1$)

**input:** None
**output:** Newborn particle set: $\{x^{(l)}_{k+1|k}, w^{(l)}_{k+1|k}\}_{l \in [L'_k+1 \ L'_k+J_{k+1}]}$

Target birth
Compute newborn particle number: $J_{k+1} \leftarrow round(\rho_T \lambda^b_{k,k+1})$
**for** $l = L'_k + 1$ **to** $L'_k + J_{k+1}$ **do**
  Compute particle position:
  $x^{(l),c}_{k+1|k} \sim \mathcal{U}([x^c_{min} \ x^c_{max}])$, $y^{(l),c}_{k+1|k} \sim \mathcal{U}([y^c_{min} \ y^c_{max}])$
  Compute particle velocity:
  $\theta \sim \mathcal{U}([-\pi \ \pi])$, $v \sim \mathcal{U}([0 \ v_{max}])$
  $\dot{x}^{(l),c}_{k+1|k} \leftarrow v\cos(\theta)$, $\dot{y}^{(l),c}_{k+1|k} \leftarrow v\sin(\theta)$
  Set particle weight: $x^{(l)}_{k+1|k} \leftarrow \frac{\lambda^b_{k,k+1}}{J_{k+1}}$
**end for**

### Sensor management

As illustrated in the PENT (figure 3.9) and BET (figure 3.15) data flows, these sensor managers are composed of the following processes:

- a predictive update step;
- a target extractor (PENT only);
- a track extractor (BET only).

The predictive equation (proposition 3.3) is essentially the combination of the construction of ideal measurements (see algorithm 2) with a data update step, described by algorithms 8 and 9 below. Thus, its implementation need not be detailed here.

The SMC implementation of the weighted peaks extractor (definition 3.2) is straightforward, the only real issue lies in the research of a global maximum at the beginning of an extraction. Indeed, if the particles have just been resampled, all the weights are equal. In this case, the extraction of a new target should be started in areas with the highest concentration of particles. This problem was solved with a grid-based approach: the state state is discretized in knots and the weight of each particle contributes to the weight of the closest knot. Because the particle from which the extraction is started is not critical, a relatively coarse grid is sufficient (respectively 200 and 10 knots in the position and velocity dimensions). Note that the target extractor may be used at any moment during an iteration, thus the time subscripts have been omitted in the pseudo-code (algorithm 5).

Note that the peak enlargement process has been discretized. The incremental step $\delta$ must be properly tuned: a smaller value increases the computational cost of the extraction process, while a larger value decreases the accuracy of the extraction (the neighborhood weight $W_n$ may significantly exceed the target weight $W_t$). Besides, the radius of the peak upon which the target extraction is based is capped by $\Delta_{max}$ in order to avoid extracted targets with excessive covariances. In the simulations presented in this chapter, these parameters are set at values $\delta = 2$, $\Delta_{max} = 200$.

The track extractor is very close to the weighted peak extractor but for the fact that:

- the former extracts as many tracks as possible while the latter extracts a number of targets fixed beforehand;
- the former provides a reduced PHD as output (see section 3.3.2).

As for algorithm 5, time subscripts are omitted for clarity's sake in algorithm 6.

**Algorithm 5** Target extractor

**input:** Copy of the current particle set: $\{x^{(l)}, w^{(l)}\}_{l \in L}$
**output:** Set of extracted targets: $\hat{X}^{WE}$

Target number and target weight
Compute estimated target number: $N \leftarrow \sum_{l=1}^{L} w^{(l)}$
Compute extracted target number: $\hat{N} \leftarrow round(N)$
Compute target weight: $W_t \leftarrow \frac{N}{\hat{N}}$

Grid initialization
Get grid knots: $\{x_g, w_g \leftarrow 0\}_{g \in G}$
**for** $l = 1$ **to** $L$ **do**
  Find closest knot: $g_c \leftarrow \arg\min_g d_{\mathcal{X}}(x_g, x^{(l)})$
  Update knot weight: $w_{g_c} \leftarrow w_{g_c} + w^{(l)}$
**end for**

Target extraction
**for** $n = 1$ **to** $\hat{N}$ **do**
  Peak extraction
  Compute heaviest knot: $g_h \leftarrow \arg\max_g w_g$
  Compute peak: $l_p = \arg\min_l d_{\mathcal{X}}(x_{g_h}, x^{(l)})$
  **for** $l = 1$ **to** $L$ **do**
    Initialize distance-to-peak vector: $D(l) \leftarrow d_{\mathcal{X}}(x^{(l)}, x^{(l_p)}$
  **end for**

  Target extraction
  Initialize distance: $\Delta \leftarrow 0$
  Initialize neighborhood weight: $W_n \leftarrow 0$
  Initialize neighborhood set: $L_n \leftarrow \emptyset$
  **while** $W_n < W_t$ **and** $\Delta < \Delta_{max}$ **do**
    Update neighborhood set: $L_n \leftarrow \{l \in L \mid D(l) \leqslant \Delta\}$
    Update neighborhood weight: $W_n \leftarrow \sum_{l \in L_n} w^{(l)}$
    Update distance: $\Delta \leftarrow \Delta + \delta$
  **end while**
  Extract new target: $\hat{x}_n \leftarrow \sum_{l \in L_n} \frac{w^{(l)}}{W_n} x^{(l)}$

  Weight removal
  Compute reduction factor: $\alpha \leftarrow \min(1, \frac{W_t}{W_n})$
  **for** $l \in L_n$ **do**
    Find closest knot: $g_c \leftarrow \arg\min_g d_{\mathcal{X}}(x_g, x^{(l)})$
    Update knot weight: $w_{g_c} \leftarrow w_{g_c} - \alpha w^{(l)}$
    Update particle weight: $w^{(l)} \leftarrow (1 - \alpha) w^{(l)}$
  **end for**
**end for**

**Algorithm 6** Track extractor

**input:** Copy of the current particle set: $\{x^{(l)}, w^{(l)}\}_{l \in L}$
**input:** Exploration zone: $\mathcal{X}_E \subseteq \mathcal{X}$
**input:** Tracking zone: $\mathcal{X}_T \subseteq \mathcal{X}$
**output:** Set of focus tracks: $T^f$
**output:** Set of non-focus tracks: $T^f$
**output:** Reduced particle set: $\{x^{(l)}, \tilde{w}^{(l)}\}_{l \in L}$

Initialization
**for** $l = 1$ **to** $L$ **do**
    Initialize reduced weight: $\tilde{w}^{(l)} \leftarrow w^{(l)}$
**end for**
Initialize grid (alg. 5)
Initialize track sets: $T^f \leftarrow \emptyset$, $T^{nf} \leftarrow \emptyset$

Track extraction
Initialize extraction flag: $f_e \leftarrow 0$
**while** $f_e = 0$ **do**
    Peak extraction
    See alg. 5: $l_p, D(:)$

    Target extraction
    Initialize distance: $\Delta \leftarrow 0$
    Initialize neighborhood weight: $W_n \leftarrow 0$
    Initialize neighborhood set: $L_n \leftarrow \emptyset$
    Initialize enlargement flag: $f_l \leftarrow 0$
    **while** $f_l = 0$ **do**
        Update neighborhood set: $L_n \leftarrow \{l \in L \mid D_l \leqslant \Delta\}$
        Update neighborhood weight: $W_n \leftarrow \sum_{l \in L_n} w^{(l)}$
        **if** $\Delta \leqslant \Delta_H$ **then**
            **if** $W_n \geqslant W_H$ **then**
                Extract track from input particle set $\{x^{(l)}, w^{(l)}\}$ (alg. 5): $\hat{x}$
                Compute reduction factor: $\alpha \leftarrow \frac{W_H}{W_n}$
                **for** $l \in L_n$ **do**
                    Update reduced weight: $\tilde{w}^{(l)} \leftarrow (1 - \alpha)\tilde{w}^{(l)}$
                **end for**
                **if** $\hat{x} \in \mathcal{X}_E \cup \mathcal{X}_T$ **then**
                    Update non-focus track set: $T^{nf} \leftarrow T^{nf} \cup \{\hat{x}\}$
                **end if**
                Weight removal (alg. 5)
                Update enlargement flag: $f_l \leftarrow 1$
            **end if**

**else if** $\Delta \geqslant \Delta_M$ **then**
**if** $W_n \geqslant W_M$ **then**
Extract track from input particle set $\{x^{(l)}, w^{(l)}\}$ (alg. 5): $\hat{x}$
Compute reduction factor: $\alpha \leftarrow \frac{W_M}{W_n}$
**if** $\hat{x} \in \mathcal{X}_T$ **then**
Update focus track set: $T^f \leftarrow T^f \cup \{\hat{x}\}$
**else**
**for** $l \in L_n$ **do**
Update reduced weight: $\tilde{w}^{(l)} \leftarrow (1-\alpha)\tilde{w}^{(l)}$
**end for**
**if** $\hat{x} \in \mathcal{X}_E$ **then**
Update non-focus track set: $T^{nf} \leftarrow T^{nf} \cup \{\hat{x}\}$
**end if**
**end if**
Weight removal (alg. 5)
Update enlargement flag: $f_l \leftarrow 1$
**end if**
**else if** $\Delta \geqslant \Delta_L$ **then**
**if** $W_n \geqslant W_L$ **then**
Extract track from input particle set $\{x^{(l)}, w^{(l)}\}$ (alg. 5): $\hat{x}$
Compute reduction factor: $\alpha \leftarrow \frac{W_L}{W_n}$
**if** $\hat{x} \in \mathcal{X}_E \cup \mathcal{X}_T$ **then**
Update focus track set: $T^f \leftarrow T^f \cup \{\hat{x}\}$
**else**
**for** $l \in L_n$ **do**
Update reduced weight: $\tilde{w}^{(l)} \leftarrow (1-\alpha)\tilde{w}^{(l)}$
**end for**
**end if**
Weight removal (alg. 5)
Update enlargement flag: $f_l \leftarrow 1$
**end if**
**else**
Update enlargement flag: $f_l \leftarrow 1$
Update extraction flag: $f_e \leftarrow 1$
**end if**
Update distance: $\Delta \leftarrow \Delta + \delta$
**end while**
**end while**

### Measurement-based birth

The birth of particles according to the target creation model only proved to be inefficent, because the number of particles around new measurement was usually too small to maintain a presence to the next iteration. Thus, it seemed necessary to add particles around the new measurements. These particles must be granted an initial weight large enough so that they do not disappear in the first resampling step following their creation, but small enough to limit the bias. The undesired consequence is that false alarm measurement will more often than not be seen as true targets, but these particles usally disappear after a few iterations if they are not confirmed by a new measurement. This solution is obviously unsatisfying and suggests that the filter is likely to fail in low SNR scenarii.

A fixed number of particles are assigned to each new measurement, this number $\rho_M$ may be chosen somewhat smaller than the number of particles per target $\rho_T$. This reduces the computational load of each measurement, and the number of particle will increase in future iterations if new measurements increase the weight in the vicinity. The new particles are spread around the new measurement in the position subspace according to the sensor parameters, and spread uniformly in the velocity subspace.

It was decided that, since only targets inside the FOV $F_u^j$ may be detected by sensor $j$, all the newborn particles created following new measurements by sensor $j$ must fall inside $F_u^j$ or be moved to the closest point inside. One may simplify the implementation, supposedly with little effect, by either discarding any particle created outside the FOV, or even ignoring the validity check. In this latter case, though, one must still check that the newborn particles belong to the state space.

### Weight update

This operation aims to approximate the data update step given by equation (2.63) with the current measurement set $Z_{k+1}$ as input. The first step is to implement the joint partitioning (definition 2.9). Note that, in the SMC framework, the cross relation (2.57) between two sensors must be adapted. Assuming that the selected control is $u \in U_{k+1}$:

$$\forall i, j \in [1\ S],\ i\mathcal{R}_u j \Leftrightarrow (\exists l \in [1\ L_{k+1|k}], x_{k|k+1}^{(l)} \in F_u^i \cap F_u^j) \tag{4.18}$$

**Algorithm 7** Measurement-based particle birth (time $k+1$)

**input:** Current measurement set: $Z_{k+1}$
**output:** Newborn particle set: $\{x_{k+1|k}^{(l)}, w_{k+1|k}^{(l)}\}_{l\in[L'_k+J_{k+1}+1\ L_{k+1|k}]}$

Particle birth
Initialize particle counter: $L \leftarrow L'_k + J_{k+1}$
**for** $j = 1$ **to** $S$ **do**
  **for** $m = 1$ **to** $m_j^{k+1}$ **do**
    Get polar coordinates of measurement: $z_{m,k+1}^j$: $(\hat{r}, \hat{\theta})$
    **for** $l = L+1$ **to** $L+\rho_M$ **do**
      Computation of new particle state
      Compute noisy polar coordinates: $r \sim \mathcal{N}(\hat{r}, (\sigma_r^j)^2)$, $\theta \sim \mathcal{N}(\hat{\theta}, (\sigma_\theta^j)^2)$
      Compute particle position: $x_{k+1|k}^{(l),c} \leftarrow x_j^c + r\cos(\theta)$, $y_{k+1|k}^{(l),c} \leftarrow y_j^c + r\sin(\theta)$
      Compute particle velocity:
      $\theta \sim \mathcal{U}([-\pi\ \pi])$, $v \sim \mathcal{U}([0\ v_{max}])$
      $\dot{x}_{k+1|k}^{(l),c} \leftarrow v\cos(\theta)$, $\dot{y}_{k+1|k}^{(l),c} \leftarrow v\sin(\theta)$
      Set particle weight: $x_{k+1|k}^{(l)} \leftarrow \frac{w_M}{\rho_M}$

      Validity check of the particle
      **if** $x_{k+1|k}^{(l)} \notin F_u^j$ **then**
        Move to closest valid state point: $x_{k+1|k}^{(l)} \leftarrow \arg\min_{x\in F_u^j} d_{\mathcal{X}}(x, x_{k+1|k}^{(l)})$
      **end if**
    **end for**
    Update particle counter: $L \leftarrow L + \rho_M$
  **end for**
**end for**
Update total number of particle number: $L_{k+1|k} \leftarrow L'_k + J_{k+1} + L$

Likewise, the target state partition (2.58) is instead a particle label partition:

$$T_k(p) = \begin{cases} \left\{ l \in [1\ L_{k+1|k}], x_{k|k+1}^{(l)} \in \overline{\bigcup_{j=1}^{S} F_k^j} \right\} & (p = 0) \\ \left\{ l \in [1\ L_{k+1|k}], x_{k|k+1}^{(l)} \in \bigcup_{j\in S_k(p)} F_k^j \right\} & (p \neq 0) \end{cases} \tag{4.19}$$

This result is interesting, because it shows that the SMC partitions are actually finer than the theoretical partitions. This suggests that the joint partitioning method is at least as efficient in the SMCPHD as in the theoretical PHD filter.

**Algorithm 8** Joint partitioning (time $k+1$)

**input:** Current multi-sensor control: $u$
**input:** Current measurement set: $Z_{k+1}$
**input:** Current particle set: $\{x^{(l)}_{k+1|k}, w^{(l)}_{k+1|k}\}_{l\in[1\ L_{k+1|k}]}$
**output:** Association matrix: $m^{k+1} \times L_{k+1|k}$ real matrix $A_u$
**output:** Missed detection matrix: $S \times L_{k+1|k}$ real matrix $D_u$
**output:** Sensor partition matrix: $P \times S$ binary matrix $S_u$
**output:** Particle partition matrix: $P \times L_{k+1|k}$ binary matrix $T_u$

Computation of association and detection matrices
**for** $m = 1$ **to** $m^{k+1}$ **do**
  Get origin sensor of measurement $z^j_{k+1}$: $j$
  **for** $l = 1$ **to** $L_{k+1|k}$ **do**
    $A_u(m,l) \leftarrow p^j_u(x^{(l)}_{k+1|k}) L^{j,z^j_{k+1}}_{k+1}(x^{(l)}_{k+1|k})$
    $D_u(j,l) \leftarrow 1 - p^j_u(x^{(l)}_{k+1|k})$
  **end for**
**end for**

Computation of adjacency matrix
Initialize $S \times S$ adjacency matrix: $A \leftarrow ((1-D_u)(1-D_u)^T > 0)$
Initialize $S \times S$ temporary matrix: $T \leftarrow (A^2 > 0)$
**while** $A \neq T$ **do**
  $A \leftarrow T$
  $T \leftarrow (A^2 > 0)$
**end while**

Joint partitioning
Initialize partition number: $p \leftarrow 0$
Initialize sensor partition matrix: $S_u \leftarrow 0$
**for** $j = 1$ **to** $S$ **do**
  **if** $A(j,:) \neq 0$ **and** $S(:,j) = 0$ **then**
    Update partition number: $p \leftarrow p + 1$
    Update sensor partition matrix: $S_u(p,:) \leftarrow A(j,:)$
  **end if**
**end for**
Compute particle partition matrix: $T_u \leftarrow (S_u(1-D_u) > 0)$

The computational cost of the partitioning is reasonable enough. The costly part is the computation of the association and detection matrices, yet these variables are required for the weight update regardless of the joint partitioning. The adjacency

matrix is a $S \times S$ binary matrix whose processing is independent of the particle number, and the only operations depending on the particle number are the products $(1 - D_u)(1 - D_u)^T > 0$ and $S_u(1 - D_u) > 0$.

The next step is the weight update using equation (2.63). The cross-terms need to be derivated in particle states $x_{k+1|k}^{(l)}$ only and can be easily built with the association $A_u$ and detection $D_u$ matrices already computed with algorithm 8. More precisely, for any multi-measurement term $C_i = \bigcup_{j \in J} \{z_{m^j}^j\}$, where $J \subseteq S_u(p)$ (see definition 2.7) and any particle $l \in T_u(p)$:

$$\beta[\delta_\emptyset, \delta_{x_{k+1|k}^{(l)}}] = \left( \prod_{j \in S_u(p)} D_u(j,l) \right) w_{k+1|k}^{(l)} \tag{4.20}$$

$$\beta[\delta_{C_i}, \delta_{x_{k+1|k}^{(l)}}] = \left( \prod_{j \in J} A_u(m^j, l) \prod_{j \in S_u(p) \setminus J} D_u(j,l) \right) w_{k+1|k}^{(l)} \tag{4.21}$$

$$\beta[\delta_{z_{m^{j_0}}^{j_0}}, 1] \simeq \sum_{l=1}^{L_{k+1|k}} \beta[\delta_{z_{m^{j_0}}^{j_0}}, \delta_{x_{k+1|k}^{(l)}}] + p_k^{fa,j_0} \qquad (C_i = \{z_{m^{j_0}}^{j_0}\}) \tag{4.22}$$

$$\beta[\delta_{C_i}, 1] \simeq \sum_{l=1}^{L_{k+1|k}} \beta[\delta_{C_i}, \delta_{x_{k+1|k}^{(l)}}] \qquad ([C_i| > 1) \tag{4.23}$$

Clearly, the computational cost comes mainly from the recursive computation of the combinational terms according to lemma 2.1, but it is quite challenging to evaluate in the general case. This will be discussed further in conclusion.

### Resampling

This operation is common in SMC methods in order to limit [Vo 05]:

- particle degeneracy, i.e. the concentration of the total weight in a small number of particles;
- a growing number of particles regardless of the estimated target number.

The resampling implemented in this thesis is very simple. It is systematic (i.e. processed at each time step) and the number of representatives (in the resampled set) of each particle follows a multinomial distribution with parameters proportional to the particle weights [Joha 05].

**Algorithm 9** Weight update (time $k+1$)

**input:** Current measurement set: $Z_{k+1}$
**input:** Time updated particle set: $\{x^{(l)}_{k+1|k}, w^{(l)}_{k+1|k}\}_{l\in[1\ L_{k+1|k}]}$
**input:** Association matrix: $A_u$
**input:** Detection matrix: $D_u$
**input:** Sensor partition matrix: $S_u$
**output:** Particle partition matrix: $T_u$
**output:** Reweighted particle set: $\{x^{(l)}_{k+1|k}, \tilde{w}^{(l)}_{k+1|k}\}_{l\in[1\ L_{k+1|k}]}$

**for** $p = 1$ **to** $P$ **do**
  Get sensors from current partition: $S_u(p) \leftarrow \{j \in S \mid S_u(p,j) = 1\}$
  Get particles from current partition: $T_u(p) \leftarrow \{l \in [1\ L_{k+1|k} \mid T_u(p,l) = 1\}$

  Computation of $\beta[\delta_\emptyset, .]$
  **for** $l \in T_u(p)$ **do**
    $\beta[\delta_\emptyset, \delta_{x^{(l)}_{k+1|k}}] \leftarrow \left(\prod_{j\in S_u(p)} D_u(j,l)\right) w^{(l)}_{k+1|k}$
  **end for**

  Computation of remaining cross-terms
  Compute combinational terms according to lemma 2.1: $\mathcal{C}(Z^{(p)}_{k+1})$
  **for** $C \in \mathcal{C}(Z^{(p)}_{k+1})$ **do**
    **for** $C_i \in C$ **do**
      Get measurements in $C_i$: $(z^j_{m^j})_{j\in J}$
      **for** $l \in T_u(p)$ **do**
        $\beta[\delta_{C_i}, \delta_{x^{(l)}_{k+1|k}}] \leftarrow \left(\prod_{j\in J} A_u(m^j, l) \prod_{j\in S_u(p)\setminus J} D_u(j,l)\right) w^{(l)}_{k+1|k}$
      **end for**
      $\beta[\delta_{C_i}, 1] \leftarrow \sum_{l\in T_u(p)} \beta[\delta_{C_i}, \delta_{x^{(l)}_{k+1|k}}]$
      **if** $J = j_0$ **then**
        $\beta[\delta_{C_i}, 1] \leftarrow \beta[\delta_{C_i}, 1] + p^{fa,j}_k$
      **end if**
    **end for**
  **end for**

  Weight update
  **for** $l \in T_u(p)$ **do**
    Update weight $w^{(l)}_{k+1|k}$ using equation (2.63)
  **end for**
**end for**

That is, the number of representatives $\zeta_{k+1}^{(l)}$ of particle $x_{k+1|k}^{(l)}$ in the resampled set is random and follows the conditions [Vo 05]:

- $\sum_{l=1}^{L_{k+1|k}} \zeta_{k+1}^{(l)} = L_{k+1|k+1}$;
- $\mathbb{E}[\zeta_{k+1}^{(l)}] = L_{k+1|k+1} \frac{\tilde{w}_{k+1|k}^{(l)}}{\sum_{l'=1}^{L_{k+1|k}} \tilde{w}_{k+1|k}^{(l')}}$.

The number of particles in the resampled set is deterministic and chosen such that the ratio of particles per target is as close as possible to the desired ratio $\rho_T$:

$$L_{k+1|k+1} = round(\rho_T \sum_{l=1}^{L_{k+1|k}} \tilde{w}_{k+1|k}^{(l)}) \tag{4.24}$$

**Algorithm 10** Resampling (time $k+1$)

**input:** Reweighted particle set: $\{x_{k+1|k}^{(l)}, \tilde{w}_{k+1|k}^{(l)}\}_{l \in [1\ L_{k+1|k}]}$
**output:** Resampled particle set: $\{x_{k+1|k+1}^{(l)}, w_{k+1|k+1}^{(l)}\}_{l \in [1\ L_{k+1|k+1}]}$

Computation of the new particle number
Compute expected target number: $N_{k+1} \leftarrow \sum_{l=1}^{L_{k+1|k}} \tilde{w}_{k+1|k}^{(l)}$
Compute new particle number: $L_{k+1|k+1} \leftarrow round(\rho_T N_{k+1})$

Resampling
**for** $l = 1$ **to** $L_{k+1|k}$ **do**
    Compute multinomial parameter: $p^{(l)} \leftarrow \frac{\tilde{w}_{k+1|k}^{(l)}}{N_{k+1}}$
**end for**
Sample $L_{k+1|k+1}$ particles from $\{x_{k+1|k}^{(l)}\}_{l=1}^{L_{k+1|k}}$ with parameters $\{p^{(l)}\}_{l=1}^{L_{k+1|k}}$: $\{x_{k+1|k+1}^{(l)}\}_{l=1}^{L_{k+1|k+1}}$
**for** $l = 1$ **to** $L_{k+1|k+1}$ **do**
    Set weight: $w_{k+1|k+1}^{(l)} \leftarrow \frac{1}{L_{k+1|k+1}}$
**end for**

## 4.3 Simulation results

This section provides the main simulation results illustrating the concepts developed in the chapters before. The different sensors in the simulations have been parametrized with various values. Their position in the surveillance region, their FOV shape and their (eventual) control constraints are critical and will be stated explicitely. The class of sensor will be also depicted with a color (green squares are $1^{st}$ class, blue squares are $2^{nd}$). The standard deviation parameters $\sigma_r^j$, $\sigma_\theta^j$ and $\sigma_{\dot{r}}^j$ are widely distributed among the sensors, spanning respectively from 2 $m$ to 10 $m$, 2° to 7° and 1 $m.s^{-1}$ to 5 $m.s^{-1}$. The detection probabilities $p^{j,d}$ and the false alarm probabilities $p^{j,fa}$ are distributed as well, spanning respectively from 0.8 to 0.98 and from $0.5 \times 10^{-6}$ to $1 \times 10^{-5}$. The ratios $\rho_T$, $\rho_M$ are fixed and equal to 100 and 50 respectively. The scenarii beginning with no targets, the filters are also initialized with no particles.

Unless otherwise stated, the figures depicting the simulation results are based on the average of several Monte Carlo runs. The target displacement patterns are deterministic and identical, each run differs from others through the random processes involved in the simulation (target detections, false alarms, true measurements, particle evolution, etc.). Whenever several methods are compared on a same scenario, the seed of the random functions in the $i$-th run of each method are initialized at the same value in order to limit bias.

### 4.3.1 Brute Force vs. Partition

The aim of this simulation is to illustrate the advantage of using the joint partitioning (theorem 2.5) rather than the "brute force" approach (theorem 2.4) for the computation of the data update step. The following results are an updated version of those in the conference paper [Dela 11b].

The surveillance region is free of geographical elements and the targets behave according to the free model (see section 4.1.1). Since this simulation emphasizes on the data update step there is no sensor management and the FOVs are fixed in the surveillance region. The target trajectories and the FOV configuration are illustrated in figure 4.7.Note that the FOV configuration is favorable for a partitioning, since the sensor partition should not be coarser than $\{1,2,3,4\} - \{5\} - \{6,7,8\} - \{9,10\}$ at any time during the simulation.

At every iteration, the data update step is computed in parallel with the brute force approach and the partition method. Then the two posterior PHDs are compared, and finally the "brute force" posterior is kept as input for the next iteration. The results were averaged on 5 Monte Carlo runs.

**Figure 4.7** Target trajectories and FOV configuration

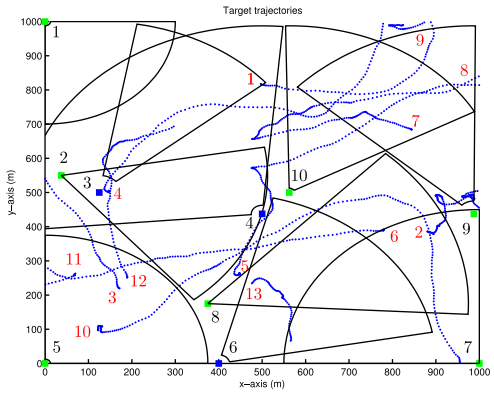



**Figure 4.8** Target trajectories (detail)

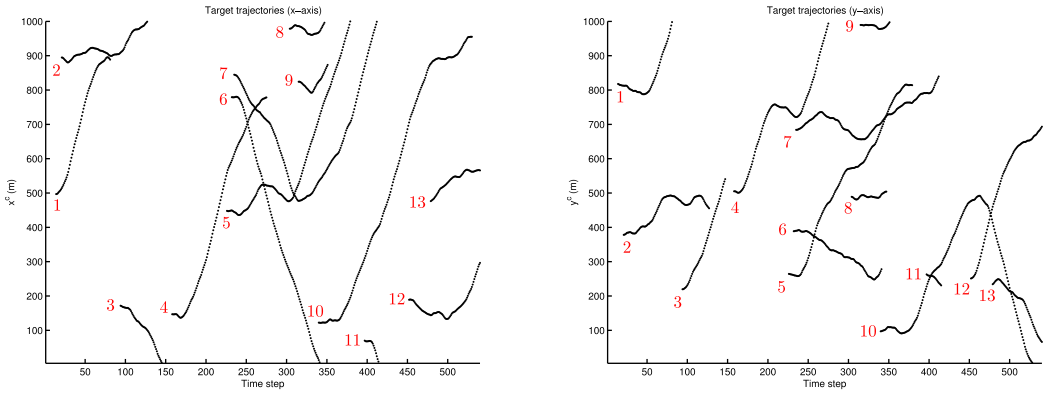



**Figure 4.9** Target number

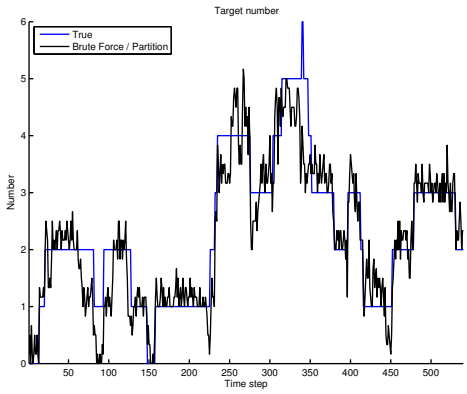


**Figure 4.10** Computing time and Kullback-Leibler divergence

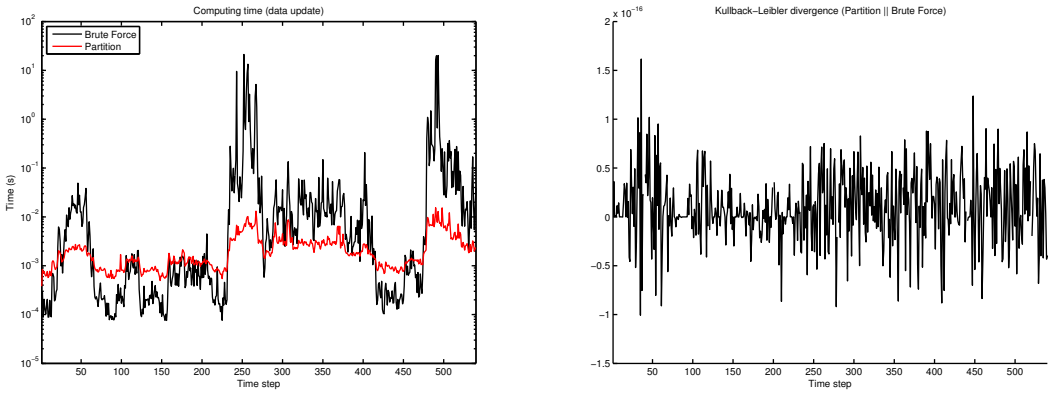



Figure 4.10 (left) shows that the computational cost of the brute force approach increases dramatically around time steps 250 and 480. Clearly, the first peak is explained by the relatively large number of targets in the surveillance region at this time (figure 4.9). The trajectories (figure 4.8) show that target 12 is in the FOV of sensors 2, 3 and 4 around time step 480. This situation is likely to produce an relatively large number of measurements, thus increasing the complexity of the data update step. As expected, the computational cost of the data update step with partitioning is significantly reduced, while the updated PHDs with the two methods remain identical (figure 4.10). Note also that the computational cost of the partition method sometimes exceeds the brute force's, typically when the target number is very low and the cost of the partitioning itself does not compensate the computational gain in the data update step. Even in these situations, however, the computational cost of the partition method remains reasonable enough.

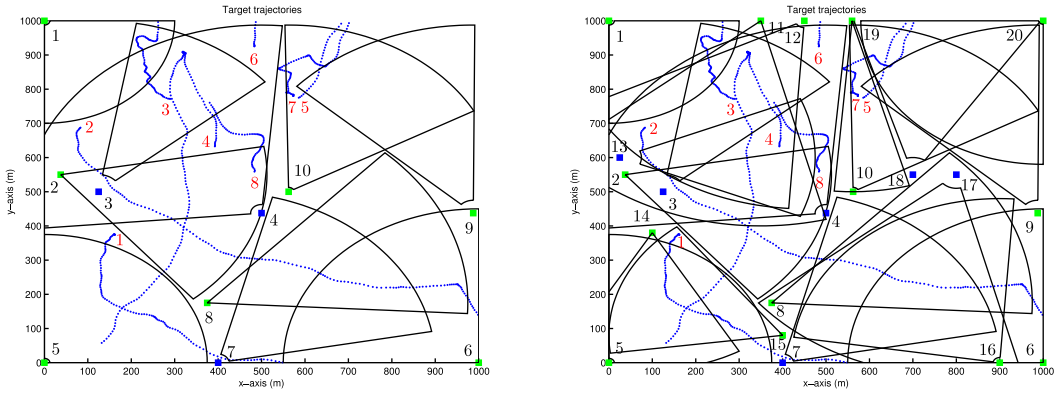
### 4.3.2 Partition method vs. ICA

It is well known that the approximation produced by the ICA (definition 2.11) depends on the sensor order, even though simulations seem to show that it does not result in noticeable differences in performance [Mahl 10a]. To the author's knowledge, the partition method (theorem 2.5) provides the first opportunity to evaluate the performance of the ICA with respect to the exact multi-sensor PHD. The following results are an updated version of those in the conference paper [Dela 11a].

The surveillance region is free of geographical elements and the targets behave according to the free model (see section 4.1.1). Since this simulation emphasizes on the data update step there is no sensor management and the FOVs are fixed in the surveillance region. As explained in the analysis of the ICA (section 2.4.2), the dis-

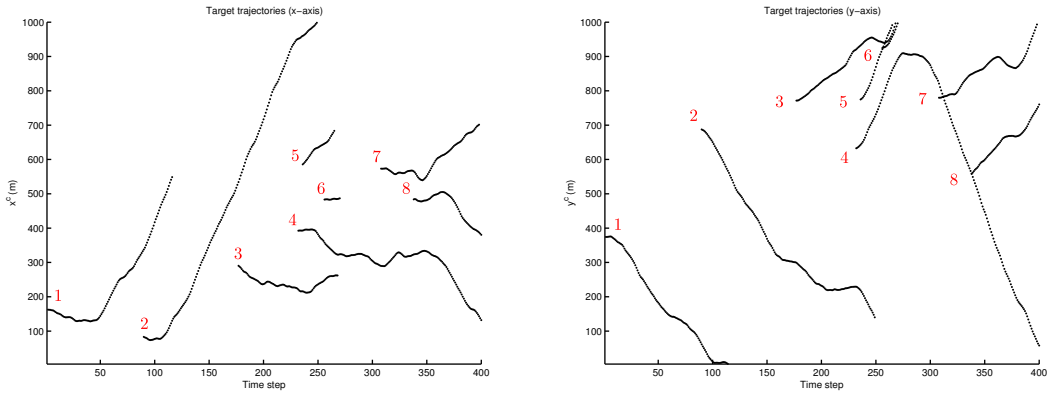
crepancies between the exact multi-sensor PHD and the approximation given by the ICA are likely to increase with the number of sensors. To illustrate this, the same scenario has been tested with two different sets of sensors:

**Figure 4.11** Target trajectories and FOV configuration




The first FOV configuration (figure 4.11, left) is identical to section 4.3.1, that is, the sensor partition should not be coarser than $\{1,2,3,4\} - \{5\} - \{6,7,8\} - \{9,10\}$ at any time during the simulation. Ten sensors have been added in the second FOV configuration (figure 4.11, right) in which the coarsest sensor partition should be $\{1,2,3,4,11,12,13\} - \{5,14,15\} - \{6,7,8,16,17\} - \{9,10,18,19,20\}$.

**Figure 4.12** Target trajectories (detail)




In order to illustrate the dependence of the ICA to the sensor order, the sensor orders that were likely to produce the best and worst estimations were estimated - the criteria being the OSPA distance between the sets of true and extracted targets averaged over the whole simulation (400 iterations) and over 20 Monte Carlo runs. Simulating the scenario with each possible sensor order was clearly out of reach in the

10-sensor configuration, let alone in the 20-sensor configuration. It is easy to prove that, provided that the sensors are partitioned according to the method explained in chapter 2, the ICA is sensitive to the sensor order *within its partition element only*. Consider for example a 4-sensor system such that, at time $k$, the sensor partition is $\{1,2\} - \{3,4\}$. Then, using the ICA with orders $1 \to 2 \to 3 \to 4$, $3 \to 4 \to 1 \to 2$, $3 \to 1 \to 2 \to 4$, etc. would produce the *exact same* posterior PHD. Thus, since the 10-sensor configuration has a coarsest partition by construction, it is sufficient to consider the permutations inside these four elements.

The resulting process being still exceedingly difficult to solve, it was approximated by decoupling the four elements from the coarset partition. First, all the permutations of $\{1,2,3,4\}$ were combined to a fixed order for the other sensors (typically $5 \to \cdots \to 10$). The 24 permutations were tested and the order that provided the best ICA was stored, and so on for the partition elements $\{6,7,8\}$ and $\{9,10\}$. The worst ICA has been approximated with the same method. This research of the best and worst ICA in the 20-sensor configuration was further simplified by keeping the best and worst order that were found in the 10-sensor configuration.

The best and worst ICA were then compared with the exact PHD (provided by the partition method) over the same 20 Monte Carlo runs.

**Figure 4.13** Target number

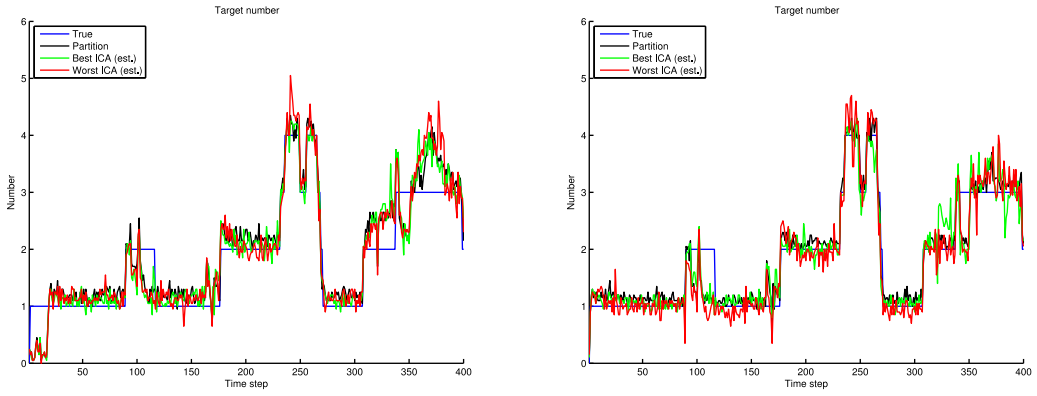
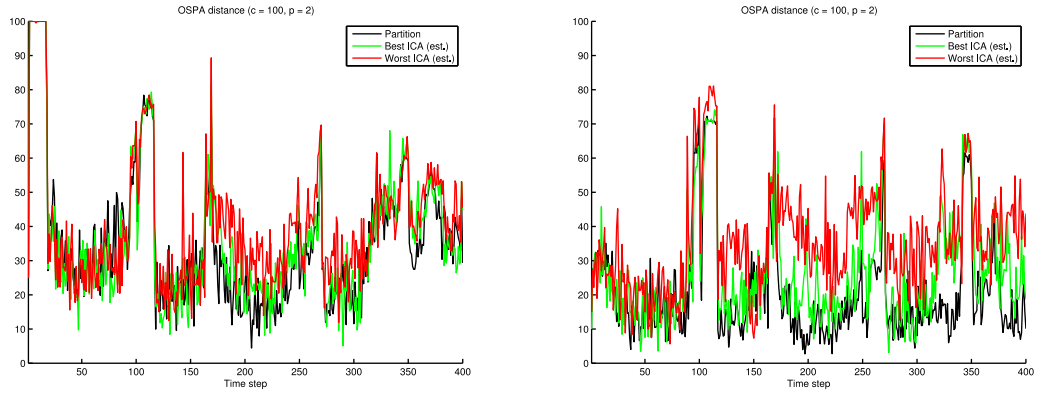



Figure 4.13 shows that, regardless of the configuration, the estimation of the target number is fairly similar with the partition method and both ICAs. The estimation of the worst ICA seems a bit more spikier than the best ICA's in the 10-sensor configuration, and the discrepancies grow larger in the 20-sensor configuration. Note that the estimation of both ICAs deteriorate in the last quarter of the simulation. The target trajectories (figure 4.12) show that this period roughly matches the life span of target 8, which is evolving in the "critical spot" of the surveillance region

where a large number of sensors are present. As expected, the quality of the ICA decreases in areas where the sensor FOVs are overlapping.

**Figure 4.14** OSPA distance ($c = 100$, $p = 2$)




The OSPA distances (figure 4.14) are more suggestive and clearly shows the discrepancies between the two ICAs, clearly growing in the 20-sensor case. The partition method provides a better estimation overall, especially during the last quarter of the simulation (confirmed by table 4.1). It also shows that, during the first half of the simulation, the ICAs are sometimes better than the partition method. This is a clear reminder that, even though the partition method is by construction the best possible method "PHD-wise", it does not necessarily implies that the estimation is better than those provided by approximation methods.

Table 4.1: Partition vs. ICA: average OSPA

| 10-sensor configuration | | | |
|---|---|---|---|
| | Partition | Best ICA (est.) | Worst ICA (est.) |
| Avg. OSPA (overall) | 35.4 | 36.3 | 40.1 |
| Avg. OSPA (last quarter) | 40.0 | 42.0 | 44.8 |
| **20-sensor configuration** | | | |
| | Partition | Best ICA (est.) | Worst ICA (est.) |
| Avg. OSPA (overall) | 20.8 | 26.2 | 36.0 |
| Avg. OSPA (last quarter) | 20.7 | 31.4 | 40.3 |

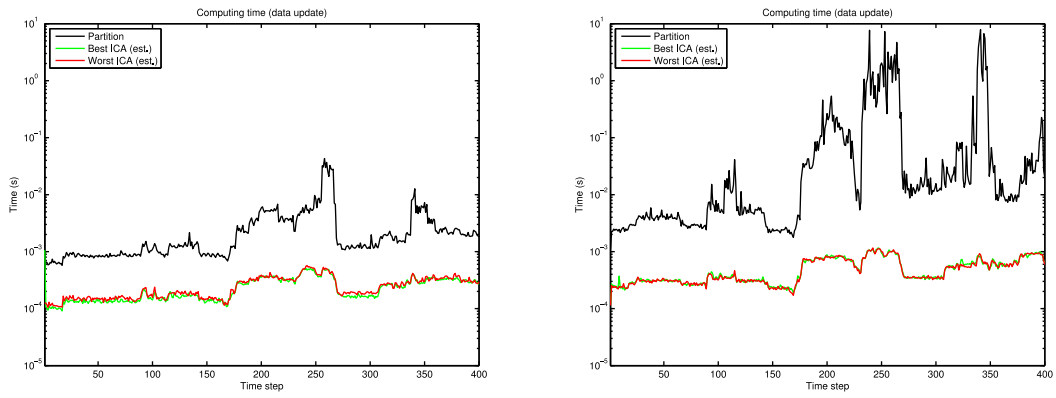
These results are hardly sufficient to draw any general conclusion on the advantage of the partition method compared to the ICA. However, they are sufficient to conclude that, on some scenarii and with some FOV configurations, the performance of the ICA depends significantly on the sensor order. This effect is obviously undesirable, all the more because there is no easy way, to the author's knowledge, to determine *a priori* the "best" sensor order. It was suggested in section 2.4.2 that ordering the sensor by increasing number of current measurements could be a sound choice. The average number of measurements per sensor and per iteration was computed in this scenario, but table 4.2 shows that there is no clear correlation between the "productivity" rank of the sensors and the orders estimated as "best" and "worst":

Table 4.2: ICA orders and average number of measurements

| 10-sensor configuration | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Criteria | Sensor rank | | | | | | | | | |
| | $1^{st}$ part. elem. | | | | $2^{nd}$ | $3^{rd}$ | | | $4^{th}$ | |
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Avg. meas. | 2 | 4 | 3 | 1 | 1 | 2 | 1 | 3 | 1 | 2 |
| ICA (best) | 4 | 1 | 3 | 2 | 1 | 2 | 1 | 3 | 1 | 2 |
| ICA (worst) | 2 | 4 | 1 | 3 | 1 | 2 | 3 | 1 | 2 | 1 |

Pay attention to the fact that the sensors are ranked in table 4.2 with respect to the order in their element of the coarsest partition. This hypothesis could be further tested by designing an ICA-based filter whose sensor order is changed *dynamically* according to the rank in productivity given by the *current* number of measurements.

**Figure 4.15** Computing time (data update)



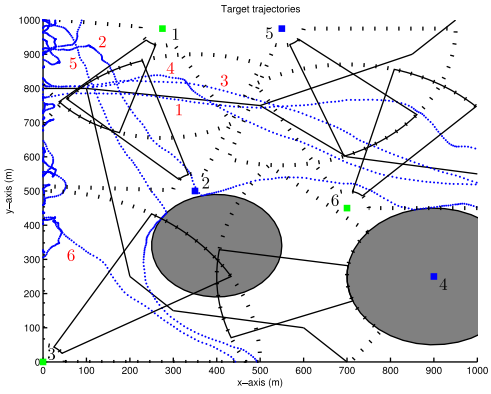
As expected, the computational cost of the partition method increases dramatically with the additional sensors. The cost is light enough in the 10-sensor configuration to consider an online implementation on a real-time tracker, but this is hardly the case in the 20-sensor configuration (figure 4.15). This result shows that, to the very least, a criteria based on the FOV configuration that would help decide *a priori* whether the partition method is tractable enough would be quite valuable. This will be discussed further in the conclusion.

### 4.3.3 PENT vs. BET

The last simulation aims at comparing the PENT and BET managers on a typical surveillance scenario. These results are presented for the first time.

The surveillance region and the target model are identical to the examples provided in the description of the ground-based model in section 4.1.1. The surveillance zone is partially covered by six sensors, some management is thus needed in order to focus the sensors on the valuable regions of the state space. The available controls are fixed for the simulation; at every time step, each sensor may be controlled according to 5 fixed directions of focus. Figure 4.16 depicts in black lines one of the five possible FOVs for each sensor, and in black dotted lines the areas that can be eventually covered by each sensor - a.k.a. the "total FOVs".

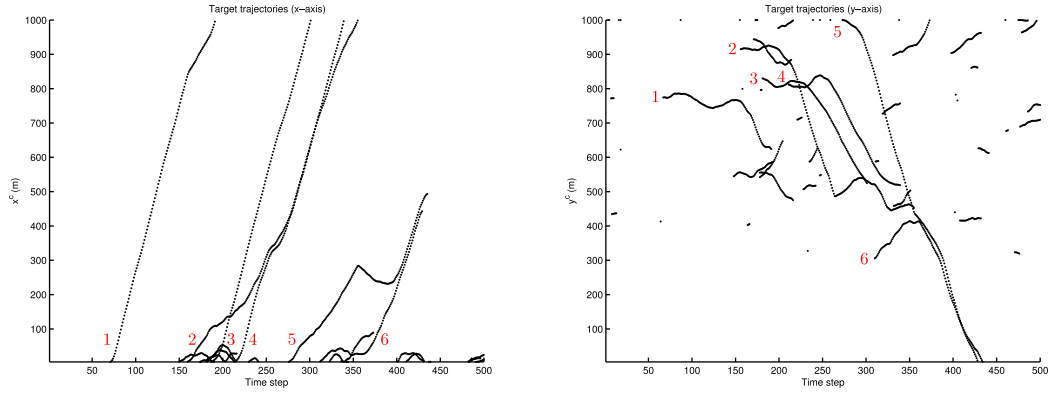
**Figure 4.16** Target trajectories and FOV configuration




Since the targets are bound to come from the left side of the surveillance region, presumably on the road, the exploration zone was set as the left part of the surveillance region (figure 4.20). Sensors 1 and 2 (figure 4.16) are mainly devoted to the exploration along the road, sensor 3 to the exploration on the lower road, sensor 4 to the tracking in the area between the obstacles, and sensor 5 and 6 to the fork in the upper road.

Note that the "total FOV" configuration is favorable to partitioning. Because the combined "total FOVs" of sensors 1 and 2, sensors 3 and 4, and sensor 5 and 6 do not overlap, the coarsest partition of the sensors should be $\{1,2\}-\{3,4\}-\{5,6\}$ at any time during the simulation. The specific configuration of the "total FOVs" simplifies the task of the sensor manager as well. Indeed, since each sensor has five possible controls, the number of available multi-sensor controls is $5^6 = 15625$. But, the total FOVs being partitioned into three elements, the independent management of partition elements $\{1,2\}$, $\{3,4\}$, $\{5,6\}$ is bound to have little effect on the management but shrinks the number of available controls to $3 \times 5^2 = 75$. This is still a lot to process for the PENT manager which requires a predictive step for every possible control (figure 3.9); thus, the predictive update equation (3.23) was approximated by six sequential updates through the single-sensor ICA (3.19) with an arbitrary sensor order. For comparison purposes, the BET manager was implemented with the ICA too. The data update step of the PHD filter (figure 1), on the other hand, was easily implemented with the exact partition method (2.63).

**Figure 4.17** Target trajectories (detail)




This scenario is quite challenging for the sensor managers. Figures 4.16 and 4.17 show that six targets are crossing the surveillance region during the simulation. Target 1 should be quite easy to follow because it stays in the vicinity of the roads. Target 2 should be more difficult to follow, notably because it changes direction in an area uncovered by the sensors. Moreover, it closely follows target 4 in the last time steps. Target 5 changes direction in an uncovered area as well, but then joins target 6 along the lower road where it should be easy to spot. Target 3 has a similar behavior as target 1 and should be quite easy to follow as well. The effet of the "indecisive" targets that enter and leave the surveillance region almost immediately should not be neglected as they are prone to disperse the focus of sensors 1, 2 and 3 at various moments during the simulation.

The critical parameters of the track extraction process (figure 3.14) are set as follows:

Table 4.3: Track parameters

| Parameter | Track level | | |
|---|---|---|---|
| | Low | Medium | High |
| Weight | 0.3 | 0.5 | 0.95 |
| Radius (m) | 160 | 120 | 40 |

The following results are averaged over 10 Monte Carlo runs. A sensor manager with a purely random strategy is added for comparison purposes.

**Figure 4.18** Target number and OSPA distance ($c = 100$, $p = 2$)

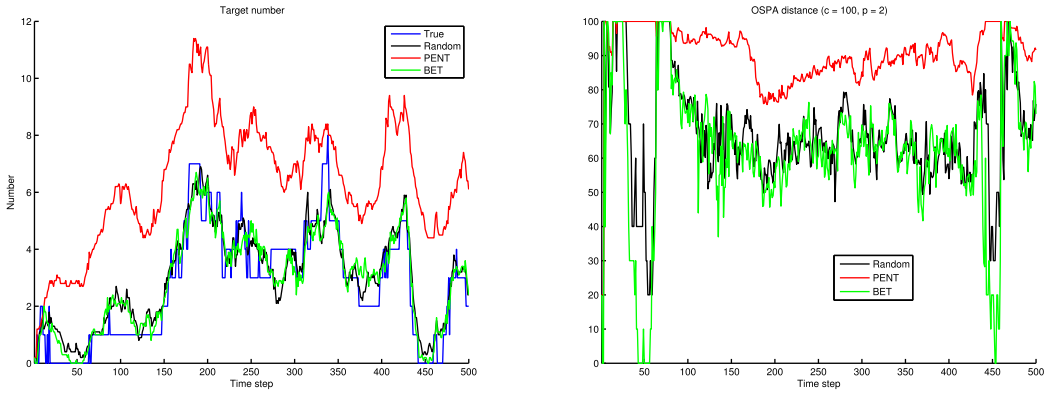



**Figure 4.19** Computing time (sensor manager)

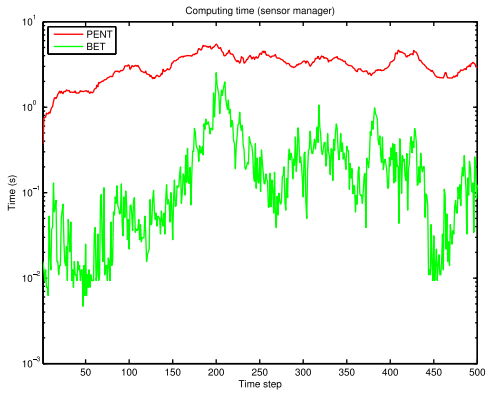


Figure 4.18 clearly shows the failure of the PENT manager on this scenario, notably because the target number is overestimated. This result is consistent with the analysis of the PENT provided in section 3.3.1 and, more precisely, in example 3.7: the sensors tend to "flee" regions with a significant amount of weight but no extracted targets in order to avoid a weight reduction in the data update. This behavior is expected and consistent with the PENT objective - increasing as much as possible the estimated number of targets, thus the total weight of the particles. Presumably, the failure of the PENT is deepened in the SMC framework. Indeed, the PENT manager leaves the regions with the highest concentration of particles unchecked where things grow out of control, especially if the particle cloud moves in an uncovered area of the state space - for example, the center of the surveillance zone in figure 4.16. Arguably, the main conclusion that can be drawn from thius failure is not the performance of the PENT itself, but the fact that the particles must be *checked periodically* by the sensors in order to avoid an explosion of the target number estimation.

Table 4.4: PENT vs. BET: average OSPA

| | PENT | Random | BET |
|---|---|---|---|
| Avg. OSPA (overall) | 90.8 | 66.3 | 61.2 |

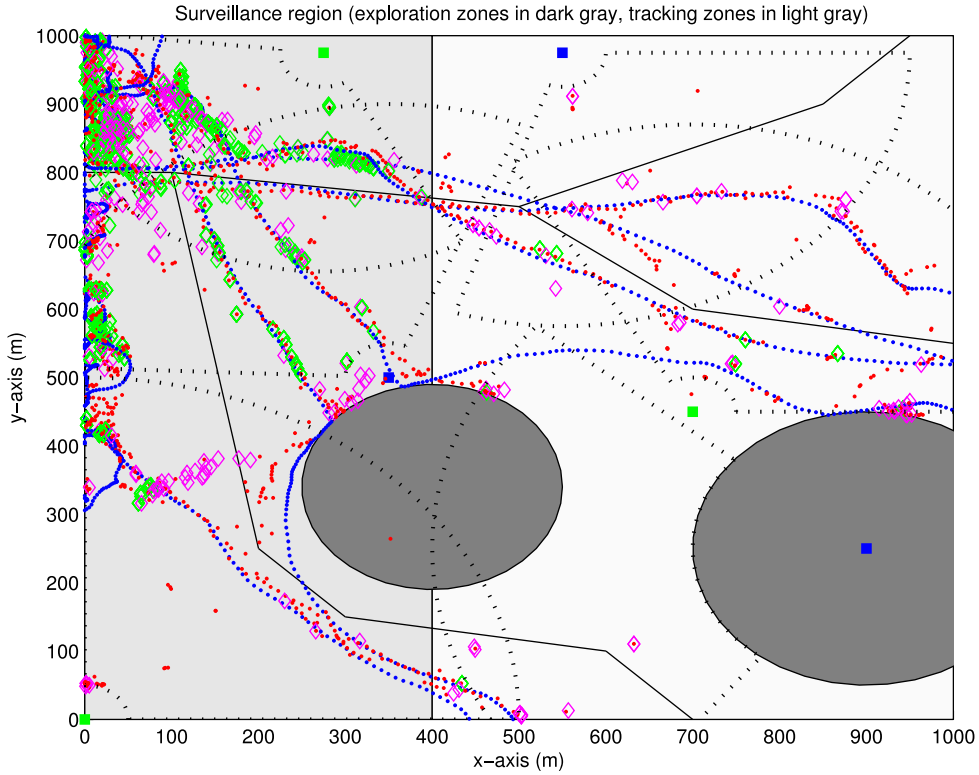
The performance of the random strategy is unexpectedly good compared to the BET manager. The estimation of the target number is fairly accurate in both cases (figure 4.18) and the OSPA error seems only slightly larger with the random method (confirmed by table 4.4). The two following points may be possible explanations of these results:

1. The first explanation pertains to the structure of this particular scenario. The FOV configuration (figure 4.16) shows that the sensors are "cramped" in a "small" surveillance region and, during the simulation, only a few directions of focus could be qualified as "bad decisions". To be sure, the performance of the PENT shows that a string of bad decisions could lead to a poor estimation; the critical advantage of the random strategy is that it guarantees at least a periodic exploration of the area covered by the "total FOVs", thus preventing the particles from growing out of control. Presumably, the difference between the random strategy and the BET manager would grow if the surveillance region were larger and contained "useless" areas never reached by the targets.

2. The second explanation is more practical and closely related to the SMC implementation. Recall from section 3.3.2 that the BET manager was designed on the principle of a periodic check of the tracks based on their level. When covered

by a sensor, a track is indeed prone to disappear if there is no new mesurements supporting the presence of a true target, and to have its level increased otherwise. Conversely, a track should "flatten" over time if unchecked and its level should decrease until it becomes a object of interest - i.e. a focus track - for the sensors. This latter principle hardly survives in the SMC implementation because unchecked particles quickly disappear in the resampling step, even if the high track level is very restrictive (see table 4.3). This phenomenon is clearly visible in the tracking zone (figure 4.20) where the presence of a non-focus track - indicating that the sensors have followed the underlying target long enough and should focus on other objects - is usually followed by an absence of extracted targets - indicating that the sensors have just stopped focusing on this point and that the particles disappeared almost immediately. This undesirable effect does not compensate a posteriori the long-term tracking of targets; to the contrary, it may overcompensate pure exploration-based strategies such as the random method. Other resampling methods are currently explored [Douc 00, Douc 05] and a modified resampling based on the effective sample size [Arul 02] has been implemented, so far to no avail.

**Figure 4.20** Illustration of the BET manager on a single run (true targets in blue, extracted targets in red, focus tracks in magenta, non-focus tracks in green)



Finally, as confirmed by figure 4.19, the BET manager is significantly lighter than the PENT, notably because it discards many superfluous predictive updates as explained in 3.3.2. An online implementation of the BET, however, is somewhat restricted to modest scenarii with a limited number of targets and, above all, a limited number of sensors.

## 4.4 Conclusion

This chapter covered the description of a few sensor and target models that allow the simulation of simple yet various surveillance scenarii. A SMC implementation of the exact multi-sensor PHD filter provided by chapter 2 and the BET manager given in chapter 3 was then proposed and thoroughly described. The advantage of the partition method was clearly shown on a first scenario, leading to the conclusion that the partitioning should always be favored over the "brute force" approach. A second scenario highlighted the influence of the sensor order on the ICA performance. The best and worst sensor orders were estimated, and the results showed that the discrepancies between the two ICAs can grow significantly with the number of sensors if the overlapping in the FOV configuration is strong enough. Besides, the approximation of both ICAs deteriorated with the increasing number of sensors compared to the exact multi-PHD fitler provided by the partition method. Finally, a third scenario showed that the BET manager can significantly outperform the PENT when the sensor coverage is limited , but also pointed out the fragility of the proposed SMC implementation.

# Conclusion and further work

## Summary

THis thesis addressed the exciting field of multi-object filtering within the PHD framework.

The first part focused on the extension of the PHD filter to the multi-sensor case. Based on previous works on the RFS theory and the single-sensor PHD, a rigorous construction of the exact multi-sensor PHD was proposed. The result was a combinatorial expression that did not provide grounds for a practical implementation. Based on the FOV configuration, a joint partitioning of the target state space and sensor indices was then proposed in order to simplify the expression of the exact multi-sensor PHD without approximation. The exact PHD was then used as a reference to compare and discuss the usual multi-sensor approximations of the PHD filter on a theoretical level.

The second part was devoted to the sensor management problem in PHD filtering. First, a rigorous extension of Mahler's PIMS to the multi-sensor case allowed the construction of the exact multi-sensor predictive PHD. Thanks to an adaptated version of the joint partitioning method, the expression of the exact predictive PHD was simplified without approximation. Then, the sensor manager introduced by Mahler - the PENT - was analyzed and its theoretical inconsistencies were shown on simple examples. Finally, the BET manager was proposed as an alternative to the PENT, but designed on more operational principles related to surveillance activities.

The last part focused on the practical implementation of the multi-sensor PHD filter and the BET manager. First, a simulation framework was built upon simple target and sensor models in order to generate various surveillance scenarii. Then, a SMC implementation of the PHD filter and the sensor manager was proposed. Finally, the main results of this thesis were illustrated on three scenarii. As expected, the first

scenario showed that the partition method reduces significantly the computational cost of the exact multi-sensor PHD, thus allowing the tractable propagation of a valuable reference PHD in scenarii with a limited number of sensors. The second scenario tested the performance of the ICA method, a well-known multi-sensor approximation. As expected, the results showed that the discrepancies between ICAs with a different sensor order in input increase with the number of sensors and are significant when the sensor number is large enough. Besides, they showed that the approximation given by the ICA detoriates, compared to the exact multi-sensor PHD, when the number of sensors increases. The third scenario aimed at comparing the PENT and BET managers on a typical surveillance scenario where the sensor coverage is limited. The BET clearly outperformed the PENT, in accuracy as well as in computational load. However, the results also pointed out the flaws of the proposed SMC implementation.

# Future work

In the scope of this thesis, the first lead that could be followed is a quantitative analysis of the computational cost of the exact multi-sensor PHD filter. It is well known that the single-sensor PHD filter has computational complexity $\mathcal{O}(m)$, where $m$ is the number of measurements, although Mahler argued [Mahl 07b] that the computational complexity of a PHD-based tracker is more accurately $\mathcal{O}(mn)$, where $n$ is the number of targets. In any case, an similar result for the exact multi-sensor PHD filter would be quite valuable because it could provide grounds for the design of hybrid filters where, for each element of the joint partitioning, one could evaluate *a priori* the computational cost of the exact data update and decide whether it is worth the trouble. The central issue in the implementation the exact data update is the computation of the combinational terms with lemma 2.1. Clearly, its computational complexity depends on the number of sensors and the number of measurements, but it also depends on the measurement distribution among the sensors - for example $|\mathcal{C}(\{z_1^1\},\{z_1^2,z_2^2,z_3^2\})| = 4$, while $|\mathcal{C}(\{z_1^1,z_2^1\},\{z_1^2,z_2^2\})| = 7$. The relation between the measurement distribution and the computational complexity is, to the author's knowledge, unknown.

Another natural lead could be the improvement of the practical implementation presented in this thesis. Arguably, the proposed SMC implementation could be enriched with an importance sampling step well-adapted to the target model. The author's understanding of the particle filtering mechanisms suggests that the critical point is the sampling of newborn particles, which must be somewhat driven by both the target birth model and the measurements. For sensor management purposes, it is indeed important to create particles in areas where targets are prone to enter in order to focus the exploration and, on the other hand, newborn particles are needed

in the vicinity of isolated measurements for the data update step to work properly [Rist 10a]. The systematic resampling step, too, would need improvement and could be replaced by other approaches [Douc 05]. Aside from the SMC framework, other implementations based on Gaussian mixtures could be envisaged. The assumption on the independence of the target detection an target survival probabilities with the target state [Pace 11] seems to preclude a direct implementation of the GMPHD in FOV-oriented sensor management problems, but this point could be further explored.

The BET manager was a first approach that leaves room for improvement, and the refinement of the track management could be the topic of future studies. Notably, in order to prevent excessively frequent changes in track levels - leading sometimes to excessively frequent changes in sensor direction of focus - a new mechanism, based on *ascending* and *descending* thresholds for each track level, could be envisaged. A *track history* must be maintained for this purpose, yet it is unavailable within the strict PHD framework. Consequently, labelisation techniques [Lin 06] must be explored beforehand. Another field of study that could be promising is the extension of the predictive PHD so that it can provides predictions several step ahead in the future, thus leading to more potent sensor managers based on lookahead policies [Bert 05]. A first solution could be built upon a sequential use of the simple predictive step, whose cost is reasonable when approximated by the ICA.

On a more theoretical level, another exciting lead for future work would be the extension of the PHD filtering principle within the RFS framework. The well-known CPHD is one such extension in which the cardinality distribution of the multi-target RFS - no longer assumed Poisson - is propagated in addition to the PHD. Another extension could be envisaged, where the second order product density would be propagated in addition to the first order product density- namely the PHD. Indeed, it seems that the factorial moments encapsulate the notion of *simultaneity* in the distribution of points, and thus the propagation of the first *and* the second order product densities could provide grounds for the design of a more complicated filter, yet able to describe *pairwise interaction* between targets while the PHD is limited to *independent* targets.

In his book [Mahl 07b] p. 595, Mahler makes an insightful remark concerning the PHD when speaking about the limited sensitivity of the ICA to the sensor order:

"This may be because the PHD approximation itself loses so much information that any information loss due to heuristic multisensor fusion is essentially irrelevant"

Aside from the fact that this thesis tried to emphasize some situations in which this information loss could hardly be qualified as irrelevant, this remark is a clear

reminder that the PHD - although a formidable tool for multi-object filtering - is first and last an approximation within the RFS framework. Arguably, the key of the PHD approximation is the Poisson assumption but, to the author's knowledge, its consequences on the validity of the PHD filter for various tracking problems is still unclear. Studying this relation is perhaps the most fundamental and challenging prospect that remains to be explored.

# Bibliography


[Aoki 11] E. Aoki, A. Begchi, P. Mandal, and Y. Boers. "A theoretical look at information-driven sensor management criteria". In: *Information Fusion, Proceedings of the 14th International Conference on*, pp. 1–8, July 2011.

[Arul 02] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. "A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking". *Signal Processing, IEEE Transactions on*, Vol. 50, pp. 174–188, 2002.

[Bert 05] D. P. Bertsekas. *Dynamic Programming and Optimal Control (Volume I)*. Vol. 1, Athena Scientific, 3 Ed., 2005.

[Blai 00] W. D. Blair and Y. Bar-Shalom, Eds. *Multitarget-Multisensor Tracking: Applications and Advances (Volume III)*. Artech House, 2000.

[Clar 06] D. E. Clark. *Multiple Target Tracking with the Probability Density Hypothesis Filter*. PhD thesis, Heriot-Watt University, Oct. 2006.

[Dela 10] E. Delande, E. Duflos, D. Heurguier, and P. Vanheeghe. "Multi-target PHD filtering: proposition of extensions to the multi-sensor case". Research Report RR-7337, INRIA, July 2010.

[Dela 11a] E. Delande, E. Duflos, D. Heurguier, and P. Vanheeghe. "Multi-sensor PHD by space partitioning: Computation of a true reference density within the PHD framework". In: *Statistical Signal Processing Workshop (SSP), 2011 IEEE*, pp. 333–336, June 2011.

[Dela 11b] E. Delande, E. Duflos, D. Heurguier, and P. Vanheeghe. "Multi-sensor PHD: Construction and implementation by space partitioning". In: *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pp. 3632–3635, May 2011.

[Douc 00] A. Doucet, S. J. Godsill, and C. Andrieu. "On sequential Monte Carlo sampling methods for Bayesian filtering". *Statistics and Computing*, Vol. 18, pp. 197–208, 2000.

[Douc 05] R. Douc, O. Cappé, and E. Moulines. "Comparison of Resampling Schemes for Particle Filtering". In: *Image and Signal Processing and Analysis, Proceedings of the 4th International Symposium on*, pp. 64–69, Sep. 2005.

[El F 08] A. El-Fallah, A. Zatezalo, R. P. S. Mahler, R. K. Mehra, and D. Donatelli. "Dynamic Sensor Management of Dispersed and Disparate Sensors for Tracking Resident Space Objects". In: *Signal Processing, Sensor Fusion, and Target Recognition XVII, Proceedings of SPIE*, 2008.

[Erdi 05] O. Erdinc, P. Willett, and Y. Bar-Shalom. "Probability Hypothesis Density Filter for Multitarget Multisensor Tracking". In: *Information Fusion, Proceedings of the 8th International Conference on*, pp. 8–8, July 2005.

[Gewe 89] J. Geweke. "Bayesian Inference in Econometric Models Using Monte Carlo Integration". *Econometrica*, Vol. 57, No. 6, pp. 1317–1339, Nov. 1989.

[Gust 02] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P.-J. Nordlund. "Particle Filter for Positioning, Navigation, and Tracking". *Signal Processing, IEEE Transactions on*, No. 2, pp. 425–437, Feb. 2002.

[Hoff 04] J. R. Hoffman and R. P. S. Mahler. "Multitarget Miss Distance via Optimal Assignment". *Systems, Man, and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, Vol. 34, No. 3, pp. 327–336, May 2004.

[Joha 05] A. M. Johansen, S. S. Singh, A. Doucet, and B.-N. Vo. "Convergence of the SMC Implementation of the PHD Filter". *Methodology and Computing in Applied Probability*, No. 2, pp. 265–291, 2005.

[Juan 09] R. R. Juang, A. Levchenko, and P. Burlina. "Tracking Cell Motion using GM-PHD". In: *Biomedical Imaging: From Nano to Macro, IEEE International Symposium on*, pp. 1154–1157, 2009.

[Kast 97] K. Kastella. "Discrimination gain to optimize detection and classification". *Systems, Man, and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, Vol. 27, No. 1, pp. 112–116, Jan. 1997.

[Kreu 04] C. M. Kreucher, K. D. Kastella, and A. O. Hero III. "Tracking Multiple Targets Using a Particle Filter Representation of the Joint Multitarget Probability Density". In: *Signal and Data Processing of Small Targets, Proceedings of SPIE*, pp. 258–269, 2004.

[Kreu 05] C. M. Kreucher, K. D. Kastella, and A. O. Hero III. "Sensor management using and active sensing approach". *Signal Processing*, Vol. 85, No. 3, pp. 607–624, March 2005.

[Lee 10] K. W. Lee, B. Kalyan, S. Wijesoma, M. Adams, F. S. Hover, and N. M. Patrikalakis. "Tracking Random Finite Objects using 3D-LIDAR in Marine Environments". In: *Proceedings of the 2010 ACM Symposium on Applied Computing*, pp. 1282–1287, 2010.

[Li R 03] X. Li Rong and V. P. Jilkov. "A Survey of maneuvering Target Tracking. Part I: Dynamic Models". *Aerospace and Electronic Systems, IEEE Transactions on*, Vol. 39, No. 4, pp. 1333–1364, Oct. 2003.

[Lin 06] L. Lin, Y. Bar-Shalom, and T. Kirubarajan. "Track labeling and PHD filter for multitarget tracking". *Aerospace and Electronic Systems, IEEE Transactions on*, Vol. 42, No. 3, pp. 778–795, July 2006.

[Liu 11] W. Liu and C. Wen. "A Linear Multisensor PHD Filter Using the Measurement Dimension Extension Approach". In: *Proceedings of the Second international conference on Advances in swarm intelligence - Volume Part II*, pp. 486–493, 2011.

[Lund 11] C. Lundquist, L. Hammarstrand, and F. Gustafsson. "Road Intensity Based Mapping Using Radar Measurements With a Probability Hypothesis Density Filter". *Signal Processing, IEEE Transactions on*, No. 4, pp. 1397–1408, 2011.

[Maeh 06] M. Maehlisch, R. Schweiger, W. Ritter, and K. Dietmayer. "Multisensor Vehicle Tracking with the Probability Hypothesis Filter". In: *Information Fusion, Proceedings of the 9th International Conference on*, pp. 1–8, July 2006.

[Mahl 02] R. P. S. Mahler. *Handbook of Multisensor Data Fusion*, Chap. 14. CRC Press, 2002.

[Mahl 03a] R. P. S. Mahler. "Multitarget Bayes Filtering via First-Order Multitarget Moments". *Aerospace and Electronic Systems, IEEE Transactions on*, Vol. 39, No. 4, pp. 1152–1178, Oct. 2003.

[Mahl 03b] R. P. S. Mahler. "Objective Functions for Bayesian Control-Theoretic Sensor Management, I: Multitarget First-Moment Approximation". In: *Aerospace Conference, IEEE Proceedings on*, pp. 1905–1923, March 2003.

[Mahl 03c] R. P. S. Mahler and T. Zajic. "Multi-Object Tracking Using a Generalized Multi-Object First-Order Moment Filter". In: *Computer Vision and Pattern Recognition Workshop, Proceedings of the 2003 Conference on*, pp. 99–99, June 2003.

[Mahl 04] R. P. S. Mahler. "Sensor Management with Non-Ideal Sensor Dynamics". In: *Information Fusion, Proceedings of the 7th International Conference on*, June 2004.

[Mahl 07a] R. P. S. Mahler. "PHD Filters of Higher Order in Target Number". *Aerospace and Electronic Systems, IEEE Transactions on*, Vol. 43, No. 4, pp. 1523–1543, Oct. 2007.

[Mahl 07b] R. P. S. Mahler. *Statistical Multisource-Multitarget Information Fusion.* Artech House, 2007.

[Mahl 07c] R. P. S. Mahler. "Unified Sensor Management Using CPHD Filters". In: *Information Fusion, Proceedings of the 10th International Conference on*, pp. 1–7, July 2007.

[Mahl 09a] R. P. S. Mahler. "The multisensor PHD Filter, I: General solution via multitarget calculus". In: *Signal Processing, Sensor Fusion, and Target Recognition XVIII, Proceedings of SPIE*, Apr. 2009.

[Mahl 09b] R. P. S. Mahler. "The multisensor PHD Filter, II: Erroneous solution via 'Poisson magic"'. In: *Signal Processing, Sensor Fusion, and Target Recognition XVIII, Proceedings of SPIE*, Apr. 2009.

[Mahl 10a] R. P. S. Mahler. "Approximate multisensor CPHD and PHD filters". In: *Information Fusion, Proceedings of the 13th International Conference on*, July 2010.

[Mahl 10b] R. P. S. Mahler. "Linear-complexity CPHD filters". In: *Information Fusion, Proceedings of the 13th International Conference on*, July 2010.

[Moya 62] J. Moyal. "The General Thory of Stochastic Population Processes". *Acta Mathematica*, Vol. 108, No. 1, pp. 1–31, Dec. 1962.

[Naga 11a] S. Nagappa and D. E. Clark. "On the ordering of the sensors in the iterated-corrector probability hypothesis density (PHD) filter". In: *Signal Processing, Sensor Fusion, and Target Recognition XVIII, Proceedings of SPIE*, Apr. 2011. http://dx.doi.org/10.1117/12.884618.

[Naga 11b] S. Nagappa, D. E. Clark, and R. P. S. Mahler. "Incorporating track uncertainty into the OSPA metric". In: *Information Fusion, Proceedings of the 14th International Conference on*, pp. 1–8, July 2011.

[Nguy 06] H. T. Nguyen. *An Introduction to Random Sets*. Chapman & Hall/CRC, 2006.

[Pace 11] M. Pace. *Stochastic models for methods for multi-object tracking*. PhD thesis, Université Bordeaux I, July 2011.

[Pant 07] K. Panta. *Multi-Target Tracking Using 1st Moment of Random Finite Sets*. PhD thesis, The University of Melbourne, Feb. 2007.

[Pant 09] K. Panta, D. E. Clark, and B.-N. Vo. "Data Association and Track Management for the Gaussian Mixture Probability Hypothesis Density Filter". *Aerospace and Electronic Systems, IEEE Transactions on*, pp. 1003–1016, Sep. 2009.

[Pham 07] N. T. Pham, W. Huang, and S. H. Ong. "Probability Hypothesis Density Approach for Multi-Camera Multi-Object Tracking". In: *Proceedings of the 8th Asian conference on Computer vision - Volume Part I*, pp. 875–884, 2007.

[Poll 09] E. Pollard, M. Rombaut, and B. Pannetier. "Algorithme hybride de pistage multicible basé sur les Ensembles Finis Aléatoires. Application à la détection de convoi de véhicules.". In: *XXIIe colloque GRETSI*, pp. –, Sep. 2009.

[Poll 10] E. Pollard. *Évolution de situations dynamiques multicibles par fusion de données spatio-temporelles*. PhD thesis, Université de Grenoble, Oct. 2010.

[Rist 10a] B. Ristic, D. E. Clark, and B.-N. Vo. "Improved SMC implementation of the PHD filter". In: *Information Fusion, Proceedings of the 13th International Conference on*, July 2010.

[Rist 10b] B. Ristic and B.-N. Vo. "Sensor Control for Multi-Object State-Space Estimation Using Random Finite Sets". *Automatica*, Vol. 46, No. 11, pp. 1812–1818, 2010.

[Rist 10c] B. Ristic, B.-N. Vo, and D. E. Clark. "Performance evaluation of multi-target tracking using the OSPA metric". In: *Information Fusion, Proceedings of the 13th International Conference on*, July 2010.

[Rist 11a] B. Ristic, B.-N. Vo, and D. E. Clark. "A note on the Reward Function for PHD Filters with Sensor Control". *Aerospace and Electronic Systems, IEEE Transactions on*, Vol. 47, No. 2, pp. 1521–1529, Apr. 2011.

[Rist 11b] B. Ristic, B.-N. Vo, D. E. Clark, and B.-T. Vo. "A Metric for Performance Evaluation of Multi-Target Tracking Algorithms". *Signal Processing, IEEE Transactions on*, Vol. 59, No. 7, pp. 3452–3457, July 2011.

[Sing 09] S. Singh, B.-N. Vo, A. Baddeley, and S. Zuyev. "Filters for Spatial Point Processes". *SIAM Journal on Control and Optimization*, Vol. 48, No. 4, pp. 2275–2295, 2009.

[Solt 11] I. Soltanzadeh, M. Azadi, and G. A. Vakili. "Using Bayesian Model Averaging (BMA) to calibrate probabilistic surface temperature forecasts over Iran". *Annales Geophysicae*, Vol. 29, No. 7, pp. 1295–1303, 2011.

[Stre 10] R. R. Streit, Ed. *Poisson Point Processes: Imaging, Tracking, and Sensing*. Springer, 2010.

[Tang 11] X. Tang and P. Wei. "Multi-target state extraction for the particle probability hypothesis density filter". *Radar, Sonar & Navigation, IET*, Vol. 5, No. 8, pp. 877–883, Oct. 2011.

[Tobi 06] M. Tobias. *Probability Hypothesis Density for Multitarget, Multisensor Tracking with Application to Passive Radar*. PhD thesis, Georgia Institute of Technology, May 2006.

[Tobi 08] M. Tobias and A. D. Lanterman. "Techniques for birth-particle placement in the probability hypothesis density filter applied to passive radar". *Radar, Sonar & Navigation, IET*, Vol. 2, No. 5, pp. 351–365, Oct. 2008.

[Ulmk 10] M. Ulmke, E. Erdinc, and P. Willett. "GMTI Tracking via the Gaussian Mixture Cardinalized Probability Hypothesis Density Filter". *Aerospace and Electronic Systems, IEEE Transactions on*, Vol. 46, pp. 1821–1833, Oct. 2010.

[Vere 88] D. Vere-Jones and D. Daley. *An Introduction to the Theory of Point Processes*. Springer Series in Statistics, 1 Ed., 1988.

[Viho 04] M. Vihola. *Random Sets for Multitarget Tracking and Data Fusion*. PhD thesis, Tampere University of Technology, Aug. 2004.

[Vo 03] B.-N. Vo, S. Singh, and A. Doucet. "Sequential Monte Carlo Implementations of the PHD Filter for Multi-target tracking". In: *Information Fusion, Proceedings of the 6h International Conference on*, pp. 792–799, 2003.

[Vo 05] B.-N. Vo, S. Singh, and A. Doucet. "Sequential Monte Carlo methods for Multi-target Filtering with Random Finite Sets". *Aerospace and Electronic Systems, IEEE Transactions on*, Vol. 41, No. 4, pp. 1224–1245, Oct. 2005.

[Vo 06] B.-N. Vo and W.-K. Ma. "The Gaussian Mixture Probability Hypothesis Density Filter". *Signal Processing, IEEE Transactions on*, Vol. 54, No. 11, pp. 4091–4104, Nov. 2006.

[Vo 07] B.-T. Vo, B.-N. Vo, and A. Cantoni. "Analytic Implementations of the Cardinalized Probability Hypothesis Density Filter". *Signal Processing, IEEE Transactions on*, Vol. 55, pp. 3553–3567, 2007.

[Vo 08] B.-T. Vo. *Random Finite Sets in Multi-Object Filtering.* PhD thesis, The University of Western Australia, Oct. 2008.

[Volt 59] V. Volterra. *Theory of Functionals and of Integral and Integro-Differential Equations.* Dover Publications, 1959.

[Wei 08a] J. Wei, X. Wang, and V. L. Syrmos. "Efficient Multi-Target Tracking with a Two-Tier Hierarchical Wireless Sensor Network". In: *Signal Processing, 9th International Conference on*, pp. 2596–2599, Oct. 2008.

[Wei 08b] J. Wei, X. Wang, and V. L. Syrmos. "Multi-Target Tracking in a Two-Tier Hierarchical Architecture". In: *Information Fusion, Proceedings of the 11th International Conference on*, pp. 1–8, 2008.

[Yell 10] P. M. Yelland. "Bayesian forecasting of parts demand". *International Journal of Forecasting*, Vol. 59, No. 2, pp. 374–396, Apr. 2010.

[Zate 08] A. Zatezalo, A. El-Fallah, R. P. S. Mahler, R. K. Mehra, and K. Pham. "Joint Search and Sensor Management for Geosynchronous Satellites". In: *Signal Processing, Sensor Fusion, and Target Recognition XVII, Proceedings of SPIE*, 2008.

# Appendix A: Mathematical proofs

## Chapter 1: Background

### Property 1.1

This proof is straightforward using the measure theoretic formulation given in [Vo 05].

*Proof.* For any $n \in [1\ N]$, let $\mathcal{T}_n$ be any subset of $\mathcal{F}(\mathcal{X}_n)$. Using equation (1.2) gives immediately:

$$\begin{aligned} P_{\prod_{n=1}^{N}\Xi_n}\left(\bigcup_{n=1}^{N}\mathcal{T}_n\right) &= \mathbb{P}\left(\left\{\left(\prod_{n=1}^{N}\Xi_n\right)(\omega_1,...,\omega_N) \in \bigcup_{n=1}^{N}\mathcal{T}_n\right\}\right) \\ &= \mathbb{P}\left(\left\{\prod_{n=1}^{N}\Xi_n(\omega_n) \in \bigcup_{n=1}^{N}\mathcal{T}_n\right\}\right) \end{aligned}$$

Since the RFSs $\Xi_n$ are independent:

$$\begin{aligned} P_{\prod_{n=1}^{N}\Xi_n}\left(\bigcup_{n=1}^{N}\mathcal{T}_n\right) &= \prod_{n=1}^{N}\mathbb{P}\left(\left\{\Xi_n(\omega) \in \bigcup_{n=1}^{N}\mathcal{T}_n\right\}\right) \\ &= \prod_{n=1}^{N}P_{\Xi_n}(\mathcal{T}_n) \end{aligned}$$

□

### Equation (1.19)

This proof is drawn from the measure theoretic formulation given in [Vo 05].

*Proof.* Let $\mathcal{T}$ be any subset of $\mathcal{F}(\mathcal{X})$. Using equation (1.17) then (1.15) gives:

$$\begin{aligned} P_\Xi(\mathcal{T}) &= \int_{\mathcal{T}} p_\Xi(X)\mu(dX) \\ &= \sum_{n=0}^{\infty}\frac{1}{n!}\int_{\chi^{-1}(\mathcal{T})\cap\mathcal{X}^n} p_\Xi(\{x_1,...,x_n\})\lambda^n(dx_1...dx_n) \end{aligned} \tag{25}$$

On the other hand, using equation (1.2) gives:

$$\begin{aligned}
P_\Xi(\mathcal{T}) &= \mathbb{P}(\{X \in \mathcal{T}\}) \\
&= \sum_{n=0}^{\infty} \rho_\Xi(n) \mathbb{P}(\{X \in \mathcal{T} \mid |X| = n\}) \\
&= \sum_{n=0}^{\infty} \rho_\Xi(n) \mathbb{P}(\{X = \chi(x_1, ...x_n) \mid (x_1, ...x_n) \in \chi^{-1}(\mathcal{T}) \cap \mathcal{X}^n\}) \\
&= \sum_{n=0}^{\infty} \rho_\Xi(n) P_\Xi^{(n)}(x_1, ...x_n \mid (x_1, ...x_n) \in \chi^{-1}(\mathcal{T}) \cap \mathcal{X}^n) \\
&= \sum_{n=0}^{\infty} \rho_\Xi(n) P_\Xi^{(n)}(\chi^{-1}(\mathcal{T}) \cap \mathcal{X}^n)
\end{aligned}$$

Which gives, using the definition of Janossy measures (1.18):

$$P_\Xi(\mathcal{T}) = \sum_{n=0}^{\infty} \frac{J_\Xi^{(n)}(\chi^{-1}(\mathcal{T}) \cap \mathcal{X}^n)}{n!}$$

Assuming that the Janossy measures admit densities:

$$\begin{aligned}
P_\Xi(\mathcal{T}) &= \sum_{n=0}^{\infty} \frac{1}{n!} \int_{\chi^{-1}(\mathcal{T}) \cap \mathcal{X}^n} j_\Xi^{(n)}(\{x_1, ..., x_n\}) dx_1 ... dx_n \\
&= \sum_{n=0}^{\infty} \frac{1}{n!} \int_{\chi^{-1}(\mathcal{T}) \cap \mathcal{X}^n} j_\Xi^{(n)}(\{x_1, ..., x_n\}) K_{\mathcal{X}}^n \lambda^n (dx_1 ... dx_n) \qquad (26)
\end{aligned}$$

Using results (25) and (26) yields:

$$p_\Xi(\{x_1, ..., x_n\}) = j_\Xi^{(n)}(x_1, ..., x_n) K_{\mathcal{X}}^n$$

□

## Property 1.2

This proof is straightforward using the definition of PGFls provided in [Vo 08]. Note that the same result is given in Moyal's earlier work on stochastic population processes (see [Moya 62] for more details).

*Proof.* Let $\Xi$ be a RFS. We have immediately:

$$\begin{aligned} G_\Xi[h] &= \mathbb{E}[h^{\Xi(\omega)}] \\ &= \sum_{n=0}^{\infty} \rho_\Xi(n) \int \cdots \int h^{\{x_1,...,x_n\}} P_\Xi^{(n)}(dx_1, ..., dx_n) \\ &= \sum_{n=0}^{\infty} \frac{1}{n!} \int \cdots \int h(x_1)...h(x_n) J_\Xi^{(n)}(dx_1, ..., dx_n) \\ &= \sum_{n=0}^{\infty} \frac{1}{n!} J_\Xi^{(n)}[h, ..., h] \end{aligned}$$

□

## Lemma 1.1

Even though the following proof was produced by the author based on the definitions and notations given in [Vo 08], a much more elegant proof is provided by Moyal in [Moya 62]. An earlier version of this result may be found in Volterra's work (see [Volt 59], p.29, for more details).

*Proof.* Let $\Xi$ be a RFS. First, let us prove by induction on $N$ that, for all $N \in \mathbb{N}$:

$$\begin{aligned} &G_\Xi^{(N)}[h + \epsilon_{N+1} g_{N+1}; g_1, ..., g_N] - G_\Xi^{(N)}[h; g_1, ..., g_N] \\ &= \sum_{n=1}^{\infty} \frac{1}{n!} \sum_{p=1}^{n} \binom{n}{p} \epsilon_{N+1}^p J_\Xi^{(n+N)}[\underbrace{h}_{n-p}, \underbrace{g_{N+1}}_{p}, g_N, ..., g_1] \end{aligned} \tag{27}$$

Definition 1.10 of functional derivatives gives the result for step 0. Assuming that (27) is true for step $N$ and using definition 1.10 again gives:

$$\begin{aligned} &G_\Xi^{(N+1)}[h + \epsilon_{N+2} g_{N+2}; g_1, ..., g_{N+1}] - G_\Xi^{(N+1)}[h; g_1, ..., g_{N+1}] \\ &= \underbrace{\lim_{\epsilon \to 0^+} \frac{1}{\epsilon} \left( G_\Xi^{(N)}[h + \epsilon_{N+2} g_{N+2} + \epsilon g_{N+1}; g_1, ..., g_N] - G_\Xi^{(N)}[h + \epsilon_{N+2} g_{N+2}; g_1, ..., g_N] \right)}_{=A} \\ &\quad - \underbrace{\lim_{\epsilon \to 0^+} \frac{1}{\epsilon} \left( G_\Xi^{(N)}[h + \epsilon g_{N+1}; g_1, ..., g_N] - G_\Xi^{(N)}[h; g_1, ..., g_N] \right)}_{=B} \end{aligned}$$

Using (27) at step $N$ gives:

$$
\begin{aligned}
A &= \lim_{\epsilon\to 0^+}\frac{1}{\epsilon}\left(\sum_{n=1}^{\infty}\frac{1}{n!}\sum_{p=1}^{n}\binom{n}{p}\epsilon^p J_\Xi^{(n+N)}[\underbrace{h+\epsilon_{N+2}g_{N+2}}_{n-p},\underbrace{g_{N+1}}_{p},g_N,...,g_1]\right)\\
&= \lim_{\epsilon\to 0^+}\left(J_\Xi^{(1+N)}[g_{N+1},...,g_1]+\sum_{n=2}^{\infty}\frac{1}{n!}\binom{n}{1}J_\Xi^{(n+N)}[\underbrace{h+\epsilon_{N+2}g_{N+2}}_{n-1},g_{N+1},g_N,...,g_1]\right.\\
&\qquad\left.\underbrace{+\sum_{n=2}^{\infty}\frac{1}{n!}\sum_{p=2}^{n}\binom{n}{p}\epsilon^{p-1}J_\Xi^{(n+N)}[\underbrace{h+\epsilon_{N+2}g_{N+2}}_{n-p},\underbrace{g_{N+1}}_{p},g_N,...,g_1]}_{=\mathcal{O}(\epsilon)}\right)\\
&= J_\Xi^{(1+N)}[g_{N+1},...,g_1]+\sum_{n=2}^{\infty}\frac{1}{(n-1)!}J_\Xi^{(n+N)}[\underbrace{h+\epsilon_{N+2}g_{N+2}}_{n-1},g_{N+1},g_N,...,g_1]
\end{aligned}
$$

Likewise:

$$
B = J_\Xi^{(1+N)}[g_{N+1},...,g_1]+\sum_{n=2}^{\infty}\frac{1}{(n-1)!}J_\Xi^{(n+N)}[\underbrace{h}_{n-1},g_{N+1},g_N,...,g_1]
$$

Thus:

$$
\begin{aligned}
&G_\Xi^{(N+1)}[h+\epsilon_{N+2}g_{N+2};g_1,...,g_{N+1}]-G_\Xi^{(N+1)}[h;g_1,...,g_{N+1}]\\
&=\sum_{n=2}^{\infty}\frac{1}{(n-1)!}J_\Xi^{(n+N)}[\underbrace{h+\epsilon_{N+2}g_{N+2}}_{n-1},g_{N+1},...,g_1]\\
&\quad-\sum_{n=2}^{\infty}\frac{1}{(n-1)!}J_\Xi^{(n+N)}[\underbrace{h}_{n-1},g_{N+1},...,g_1]\\
&=\sum_{n=2}^{\infty}\frac{1}{(n-1)!}\sum_{p=0}^{n-1}\binom{n-1}{p}J_\Xi^{(n+N)}[\underbrace{h}_{n-1-p},\underbrace{\epsilon_{N+2}g_{N+2}}_{p},g_{N+1},...g_1]\\
&\quad-\sum_{n=2}^{\infty}\frac{1}{(n-1)!}J_\Xi^{(n+N)}[\underbrace{h}_{n-1},g_{N+1},...,g_1]\\
&=\sum_{n=2}^{\infty}\frac{1}{(n-1)!}\sum_{p=1}^{n-1}\binom{n-1}{p}J_\Xi^{(n+N)}[\underbrace{h}_{n-1-p},\underbrace{\epsilon_{N+2}g_{N+2}}_{p},g_{N+1},...,g_1]\\
&=\sum_{n=1}^{\infty}\frac{1}{n!}\sum_{p=1}^{n}\binom{n}{p}\epsilon_{N+2}^p J_\Xi^{(n+N+1)}[\underbrace{h}_{n-p},\underbrace{g_{N+2}}_{p},g_{N+1},...,g_1]
\end{aligned}
$$

Thus, (27) is true for all $N$. Now, using definition 1.10 and (27) gives, for any $N \in \mathbb{N}$:

$$
\begin{aligned}
&G_{\Xi}^{(N)}[h; g_1, ..., g_N] \\
&= \lim_{\epsilon \to 0^+} \epsilon \left( \sum_{n=1}^{\infty} \frac{1}{n!} \sum_{p=1}^{n} \binom{n}{p} \epsilon^p J_{\Xi}^{(n+N-1)}[\underbrace{h}_{n-p}, \underbrace{g_N}_{p}, g_{N-1}, ..., g_1] \right) \\
&= \lim_{\epsilon \to 0^+} \Bigg( J_{\Xi}^{(N)}[g_N, ..., g_1] + \sum_{n=2}^{\infty} \frac{1}{n!} \binom{n}{1} J_{\Xi}^{(n+N-1)}[\underbrace{h}_{n-1}, g_N, ..., g_1] \\
&\qquad + \underbrace{\sum_{n=2}^{\infty} \frac{1}{n!} \sum_{p=2}^{n} \binom{n}{p} \epsilon^{p-1} J_{\Xi}^{(n+N-1)}[\underbrace{h}_{n-p}, \underbrace{g_N}_{p}, g_{N-1}, ..., g_1]}_{=\mathcal{O}(\epsilon)} \Bigg) \\
&= J_{\Xi}^{(N)}[g_N, ..., g_1] + \sum_{n=1}^{\infty} \frac{1}{n!} J_{\Xi}^{(n+N)}[\underbrace{h}_{n}, g_N, ..., g_1] \\
&= \sum_{n=0}^{\infty} \frac{1}{n!} J_{\Xi}^{(n+N)}[\underbrace{h}_{n}, g_N, ..., g_1]
\end{aligned}
$$

□

## Property 1.4

This proof is adapted from Moyal's early work on stochastic population processes (see [Moya 62] for more details).

*Proof.* Using lemma 1.1 and setting $h = 0$ yields immediately:

$$G_{\Xi}^{(N)}[0; g_1, ..., g_N] = J_{\Xi}^{(N)}[g_1, ..., g_N]$$

Then, using lemma 1.1 and setting $h = 1$ gives:

$$
\begin{aligned}
G_{\Xi}^{(N)}[1; g_1, ..., g_N] &= \sum_{n=0}^{\infty} \frac{1}{n!} J_{\Xi}^{(N+n)}[g_1, ..., g_N, \underbrace{1}_{n}] \\
&= \sum_{n=0}^{\infty} \frac{1}{n!} \underbrace{\int \cdots \int}_{N+n} g_1(x_1)...g_N(x_N) 1(x_{N+1})...1(x_{N+n}) J_{\Xi}^{(N+n)}(dx_1, ..., dx_{N+n}) \\
&= \underbrace{\int \cdots \int}_{N} g_1(x_1)...g_N(x_N) \left( \sum_{n=0}^{\infty} \frac{1}{n!} \underbrace{\int \cdots \int}_{n} J_{\Xi}^{(N+n)}(dx_1, ..., dx_{N+n}) \right)
\end{aligned}
$$

Which gives, using equation (1.24):

$$G_{\Xi}^{(N)}[1; g_1, ..., g_N] = \underbrace{\int \cdots \int}_{N} g_1(x_1)...g_N(x_N)V_{\Xi}^{(N)}(dx_1 \times ... \times dx_N)$$
$$= V_{\Xi}^{(N)}[g_1, ..., g_N]$$

□

## Property 1.5

This proof is given by Mahler in [Mahl 03a].

*Proof.* Assuming $G[h] = h(x_0)$ and using the definition 1.10 of a functional derivative gives immediately:

$$\begin{aligned}\frac{\delta G}{\delta x}[h] &= \lim_{\epsilon \to 0^+} \frac{G[h + \epsilon\delta_x K_{\mathcal{X}}] - G[h]}{\epsilon} \\ &= \lim_{\epsilon \to 0^+} \frac{h(x_0) + \epsilon\delta_x(x_0)K_{\mathcal{X}} - h(x_0)}{\epsilon} \\ &= \delta_x(x_0)K_{\mathcal{X}}\end{aligned}$$

Likewise, assuming $G[h] = \int h(x)p(x)dx$ and using the definition 1.10 of a functional derivative gives immediately:

$$\begin{aligned}\frac{\delta G}{\delta x}[h] &= \lim_{\epsilon \to 0^+} \frac{G[h + \epsilon\delta_x K_{\mathcal{X}}] - G[h]}{\epsilon} \\ &= \lim_{\epsilon \to 0^+} \frac{\int_{\mathcal{X}}(h(y) + \epsilon\delta_x(y)K_{\mathcal{X}})p(y)dy - \int_{\mathcal{X}} h(y)p(y)dy}{\epsilon} \\ &= \lim_{\epsilon \to 0^+} \frac{\int_{\mathcal{X}} \epsilon\delta_x(y)K_{\mathcal{X}}p(y)\lambda(dy)}{\epsilon} \\ &= p(x)K_{\mathcal{X}}\end{aligned}$$

□

## Property 1.6

This proof is given by Mahler in [Mahl 03a]. Note that the equivalent result with the point process formulation was given earlier in [Vere 88] (see "Campbell theorem", eq. (6.4.11) p. 188).

*Proof.* We have immediately:

$$\begin{aligned}
\int_{\mathcal{F}(\mathcal{X})} h_X p_\Xi(X)\mu(dX) &= \sum_{n=1}^{\infty} \frac{1}{n!} \int_{\mathcal{X}^n} \left( \sum_{i=1}^{n} h(x_i) \right) j_\Xi^{(n)}(x_1, ..., x_n) dx_1 ... dx_n \\
&= \sum_{n=1}^{\infty} \frac{1}{n!} \sum_{i=1}^{n} \int_{\mathcal{X}^n} h(x_i) j_\Xi^{(n)}(x_1, ..., x_n) dx_1 ... dx_n \\
&= \sum_{n=1}^{\infty} \frac{1}{(n-1)!} \int_{\mathcal{X}^n} h(x_1) j_\Xi^{(n)}(x_1, ..., x_n) dx_1 ... dx_n \\
&= \int_{\mathcal{X}} h(x_1) \left( \sum_{n=1}^{\infty} \frac{1}{(n-1)!} \int_{\mathcal{X}^{n-1}} j_\Xi^{(n)}(x_1, ..., x_n) dx_2 ... dx_n \right) dx_1 \\
&= \int_{\mathcal{X}} h(x) \left( \sum_{n=0}^{\infty} \frac{1}{n!} \int_{\mathcal{X}^n} j_\Xi^{(n+1)}(x, x_1, ..., x_n) dx_1 ... dx_n \right) dx \\
&= \int_{\mathcal{X}} h(x) v_\Xi(x) dx
\end{aligned}$$

□

## Property 1.7

This proof is given by Mahler in [Mahl 03a].

*Proof.* Let $x_0$ be an arbitrary point in $\mathcal{X}$. Using equation (1.44), we can write:

$$\begin{aligned}
v_{\Xi_\Phi}(x_0) K_{\mathcal{X}} &= \left[ \frac{\delta}{\delta x_0} G_{\Xi_\Phi}[h] \right]_{h=1} \\
&= \left[ \frac{\delta}{\delta x_0} G_\Xi[\Phi[h]] \right]_{h=1} \\
&= \int_{\mathcal{F}(\mathcal{X})} \left[ \frac{\delta}{\delta x_0} \left( \Phi[h]^X \right) \right]_{h=1} p_\Xi(X)\mu(dX) \\
&= \int_{\mathcal{F}(\mathcal{X})} \left[ \sum_{x \in X} \left( \frac{\delta}{\delta x_0} \Phi[h](x) \right) \Phi[h]^{X \setminus \{x\}} \right]_{h=1} p_\Xi(X)\mu(dX) \\
&= \int_{\mathcal{F}(\mathcal{X})} \left[ \sum_{x \in X} \left( \frac{\delta}{\delta x_0} G_{\Xi_{\Phi,x}}[h] \right) \Phi[h]^{X \setminus \{x\}} \right]_{h=1} p_\Xi(X)\mu(dX) \\
&= \int_{\mathcal{F}(\mathcal{X})} \left( \sum_{x \in X} \left[ \frac{\delta}{\delta x_0} G_{\Xi_{\Phi,x}}[h] \right]_{h=1} \right) p_\Xi(X)\mu(dX)
\end{aligned}$$

That is, using equation (1.44) again:

$$\begin{aligned} v_{\Xi_\Phi}(x_0)K_{\mathcal{X}} &= \int_{\mathcal{F}(\mathcal{X})} \left( \sum_{x \in X} v_{\Xi_{\Phi,x}}(x_0)K_{\mathcal{X}} \right) p_{\Xi}(X)\mu(dX) \\ &= \int_{\mathcal{F}(\mathcal{X})} \left( \sum_{x \in X} h_{x_0}(x) \right) p_{\Xi}(X)\mu(dX) \end{aligned}$$

where $h_{x_0} : x \mapsto v_{\Xi_{\Phi,x}}(x_0)K_{\mathcal{X}}$. Then, equation (1.49) yields:

$$\begin{aligned} v_{\Xi_\Phi}(x_0)K_{\mathcal{X}} &= \int_{\mathcal{X}} h_{x_0}(x)v_{\Xi}(x)dx \\ &= \int_{\mathcal{X}} v_{\Xi_{\Phi,x}}(x_0)K_{\mathcal{X}}v_{\Xi}(x)dx \end{aligned}$$

Thus:

$$v_{\Xi_\Phi}(x_0) = \int_{\mathcal{X}} v_{\Xi_{\Phi,x}}(x_0)v_{\Xi}(x)dx$$

□

# Chapter 2: The multi-sensor PHD

## Property 2.1

This proof is drawn from Mahler's study of the Poisson RFS in [Mahl 03a].

*Proof.* Let $Y = \{y_1, ..., y_n\}$ be any set of points in $\mathcal{X}$, and $\{T_i\}_{i\in[1\ n]}$ a family of subsets of $\mathcal{X}$. Then, using the definition of Janossy measures (1.18):

$$J_\Xi^{(n)}(T_1 \times ... \times T_n) = n!\rho_\Xi(n)P_\Xi^{(n)}(T_1 \times ... \times T_n)$$

Since $\Xi$ is Poisson with parameter $\lambda_\Xi$ and spatial intensity $I_\Xi$:

$$\begin{aligned} J_\Xi^{(n)}(T_1 \times ... \times T_n) &= n!e^{-\lambda_\Xi}\frac{\lambda_\Xi^n}{n!}\int_{T_1}\dots\int_{T_n}\prod_{i=1}^{n}\frac{I_\Xi(x_i)}{\lambda_\Xi}dx_1...dx_n \\ &= e^{-\lambda_\Xi}\int_{T_1}\dots\int_{T_n}\prod_{i=1}^{n}I_\Xi(x_i)dx_1...dx_n \end{aligned} \tag{28}$$

Thus:

$$j_\Xi^{(n)}(y_1, ..., y_n) = e^{-\lambda_\Xi}\prod_{i=1}^{n}I_\Xi(y_i)$$

Then, using (1.24) gives:

$$\begin{aligned} v_\Xi^{(n)}(y_1, ..., y_n) &= \sum_{m=0}^{\infty}\frac{1}{m!}\int_{\mathcal{X}^m} j_\Xi^{(m+n)}(y_1, ..y_n, x_{n+1}, ..., x_{n+m})dx_{n+1}...dx_{n+m} \\ &= \sum_{m=0}^{\infty}\frac{1}{m!}\int_{\mathcal{X}^m}\left(\prod_{i=1}^{n}I_\Xi(y_i)\right)j_\Xi^{(m)}(x_{n+1}, ..., x_{n+m})dx_{n+1}...dx_{n+m} \\ &= \left(\prod_{i=1}^{n}I_\Xi(y_i)\right)\underbrace{\sum_{m=0}^{\infty}\frac{1}{m!}\int_{\mathcal{X}^m} j_\Xi^{(m)}(x_{n+1}, ..., x_{n+m})dx_{n+1}...dx_{n+m}}_{=1} \\ &= \prod_{i=1}^{n}I_\Xi(y_i) \end{aligned}$$

Besides, the characterization of the PGFl (1.33) gives:

$$G_\Xi[h] = \sum_{m=0}^{\infty}\frac{1}{m!}J_\Xi^{(m)}[h, ..., h]$$

That is, using (28):

$$\begin{aligned} G_\Xi[h] &= \sum_{m=0}^{\infty} \frac{1}{m!} e^{-\lambda_\Xi} \int_{\mathcal{X}^m} \prod_{i=1}^{m} h(x_i) I_\Xi(x_i) dx_1 ... dx_m \\ &= e^{-\lambda_\Xi} \sum_{m=0}^{\infty} \frac{1}{m!} \left( \int_{\mathcal{X}} h(x) I_\Xi(x) dx \right)^m \\ &= e^{-\lambda_\Xi} \sum_{m=0}^{\infty} \frac{1}{m!} (I_\Xi[h])^m \\ &= e^{I_\Xi[h] - \lambda_\Xi} \end{aligned}$$

□

## Property 2.2

This proof is an adaptation of Mahler's study of the Poisson RFS in [Mahl 03a].

*Proof.* Let $Y = \{y_1, ..., y_n\}$ be any set of points in $\mathcal{X}$, and $\{T_i\}_{i \in [1\ n]}$ a family of subsets of $\mathcal{X}$. Then, using the definition of Janossy measures (1.18):

$$J_\Xi^{(n)}(T_1 \times ... \times T_n) = n! \rho_\Xi(n) P_\Xi^{(n)}(T_1 \times ... \times T_n)$$

Since $\Xi$ is i.i.d cluster with mean $\lambda_\Xi$ and spatial intensity $I_\Xi$:

$$\begin{aligned} J_\Xi^{(n)}(T_1 \times ... \times T_n) &= n! \rho_\Xi(n) \int_{T_1} \dots \int_{T_n} \prod_{i=1}^{n} \frac{I_\Xi(x_i)}{\lambda_\Xi} dx_1 ... dx_n \\ &= \frac{n! \rho_\Xi(n)}{\lambda_\Xi^n} \int_{T_1} \dots \int_{T_n} \prod_{i=1}^{n} I_\Xi(x_i) dx_1 ... dx_n \end{aligned} \tag{29}$$

Thus:

$$j_\Xi^{(n)}(y_1, ..., y_n) = \frac{n! \rho_\Xi(n)}{\lambda_\Xi^n} \prod_{i=1}^{n} I_\Xi(y_i)$$

Besides, the characterization of the PGFl (1.33) gives:

$$G_\Xi[h] = \sum_{m=0}^{\infty} \frac{1}{m!} J_\Xi^{(m)}[h, ..., h]$$

That is, using (29):

$$\begin{aligned} G_\Xi[h] &= \sum_{m=0}^{\infty} \frac{1}{m!} \frac{m!\rho_\Xi(m)}{\lambda_\Xi^m} \int_{\mathcal{X}^m} \prod_{i=1}^{m} h(x_i) I_\Xi(x_i) dx_1 ... dx_m \\ &= \sum_{m=0}^{\infty} \rho_\Xi(m) \left( \int_{\mathcal{X}} \frac{h(x) I_\Xi(x)}{\lambda_\Xi} dx \right)^m \\ &= \sum_{m=0}^{\infty} \rho_\Xi(m) \left( \frac{I_\Xi[h]}{\lambda_\Xi} \right)^m \\ &= G_{|\Xi|} \left( \frac{I_\Xi[h]}{\lambda_\Xi} \right) \end{aligned}$$

□

## Property 2.3

This proof is drawn from Mahler's construction of the single-sensor PHD filter in [Mahl 03a].

*Proof.* Let $Y = \{y_1, ..., y_n\}$ be any set of points in $\mathcal{X}$, and $\{T_i\}_{i \in [1\ n]}$ a family of subsets of $\mathcal{X}$. Then, using the definition of Janossy measures (1.18):

$$J_\Xi^{(n)}(T_1 \times ... \times T_n) = n! \rho_\Xi(n) P_\Xi^{(n)}(T_1 \times ... \times T_n)$$

Since $\Xi$ is Bernoulli with parameter $b_\Xi$ and spatial dsitribution $I_\Xi$:

$$J_\Xi^{(n)}(T_1 \times ... \times T_n) = \begin{cases} 1 - b_\Xi & n = 0 \\ b_\Xi \displaystyle\int_{T_1} I_\Xi(x_1) dx_1 & n = 1 \\ 0 & otherwise \end{cases}$$

Thus:

$$j_\Xi^{(n)}(y_1, ..., y_n) = \begin{cases} 1 - b_\Xi & n = 0 \\ b_\Xi I_\Xi(y_1) & n = 1 \\ 0 & otherwise \end{cases}$$

Besides, the characterization of the PGFl (1.33) gives:

$$G_\Xi[h] = \sum_{m=0}^{\infty} \frac{1}{m!} J_\Xi^{(m)}[h, ..., h]$$

Thus:

$$\begin{aligned} G_{\Xi}[h] &= 1 - b_{\Xi} + b_{\Xi} \int_{\mathcal{X}} h(x) I_{\Xi}(x) dx \\ &= 1 - b_{\Xi} + b_{\Xi} I_{\Xi}[.] \end{aligned}$$

□

## Proposition 2.1

This proof is drawn from Mahler's construction of the single-sensor PHD filter in [Mahl 03a].

*Proof.* Let $\Xi^E_{k,k+1}(X)$ be the Multi-Bernoulli RFS $\bigcup_{x \in X} \Xi^E_{k,k+1}(x)$. Then, using proposition 2.4 on Multi-Bernoulli RFS gives:

$$\begin{aligned} G_{\Xi^E_{k,k+1}(X)}[h] &= \prod_{x \in X} \left( (1 - p^s_{k,k+1}(x) + p^s_{k,k+1}(x) p^t_{k,k+1}[h|x]) \right) \\ &= (1 - p^s_{k,k+1}(.) + p^s_{k,k+1}(.) p^t_{k,k+1}[h|.])^X \end{aligned}$$

Let $\Xi^S_{k,k+1}(X)$ be the union RFS $\bigcup_{x \in X} \Xi^S_{k,k+1}(x)$. Since the spawning RFS are independent, using property 1.3 on union RFS yields:

$$\begin{aligned} G_{\Xi^S_{k,k+1}(X)}[h] &= \prod_{x \in X} G_{\Xi^S_{k,k+1}(x)}[h] \\ &= (G_{\Xi^S_{k,k+1}(.)}[h])^X \end{aligned}$$

By construction, the transition RFS $\Xi^T_{k,k+1}(X)$ is the union RFS of independent RFS $\Xi^E_{k,k+1}(X)$, $\Xi^S_{k,k+1}(X)$ and $\Xi^B_{k,k+1}$. Thus, using property 1.3 gives:

$$\begin{aligned} G_{\Xi^T_{k,k+1}(X)}[h] &= G_{\Xi^E_{k,k+1}(X)}[h] G_{\Xi^E_{k,k+1}(X)}[h] G_{\Xi^B_{k,k+1}}[h] \\ &= (1 - p^s_{k,k+1}(.) + p^s_{k,k+1}(.) p^t_{k,k+1}[h|.])^X (G_{\Xi^S_{k,k+1}(.)}[h])^X G_{\Xi^B_{k,k+1}}[h] \end{aligned}$$

□

## Theorem 2.1

This is proof is a key element from Mahler's construction of the single-sensor PHD filter (see [Mahl 03a]). An earlier account of a similar construction may be found in Moyal's work (see [Moya 62] for more details).

*Proof.* Let $h$ be any real-valued function in [0 1]. Using the definition of the PGFl (1.29) on predicted RFS $\Xi_{k+1|k}$ gives:

$$G_{\Xi_{k+1|k}}[h] = \int_{\mathcal{F}(\mathcal{X})} h^Y p_{\Xi_{k+1|k}}(Y|Z_{1:k})\mu(dY)$$

That is, thanks to the RFS filter equation (1.55):

$$G_{\Xi_{k+1|k}}[h] = \int_{\mathcal{F}(\mathcal{X})} h^Y \left( \int_{\mathcal{F}(\mathcal{X})} p_{\Xi^T_{k,k+1}}(Y|X) p_{\Xi_{k|k}}(X|Z_{1:k})\mu(dX) \right) \mu(dY)$$
$$= \int_{\mathcal{F}(\mathcal{X})} \left( \int_{\mathcal{F}(\mathcal{X})} h^Y p_{\Xi^T_{k,k+1}}(Y|X)\mu(dY) \right) p_{\Xi_{k|k}}(X|Z_{1:k})\mu(dX)$$

Using the definition of the PGFl (1.29) again gives:

$$G_{\Xi_{k+1|k}}[h] = \int_{\mathcal{F}(\mathcal{X})} G_{\Xi^T_{k,k+1}(X)}[h] p_{\Xi_{k|k}}(X|Z_{1:k})\mu(dX)$$

With the assumptions of proposition 2.1:

$$G_{\Xi_{k+1|k}}[h]$$
$$= \int_{\mathcal{F}(\mathcal{X})} (1 - p^s_{k,k+1}(.) + p^s_{k,k+1}(.) f^t_{k,k+1}[h|.])^X (G_{\Xi^S_{k,k+1}(.)}[h])^X G_{\Xi^B_{k,k+1}}[h] p_{\Xi_{k|k}}(X|Z_{1:k})\mu(dX)$$
$$= G_{\Xi^B_{k,k+1}}[h] \int_{\mathcal{F}(\mathcal{X})} \left( (1 - p^s_{k,k+1}(.) + p^s_{k,k+1}(.) f^t_{k,k+1}[h|.]) G_{\Xi^S_{k,k+1}(.)}[h] \right)^X p_{\Xi_{k|k}}(X|Z_{1:k})\mu(dX)$$

Let $\Phi : h \mapsto \Phi[h]$ be such that $\Phi[h](.) = (1 - p^s_{k,k+1}(.) + p^s_{k,k+1}(.) f^t_{k,k+1}[h|.]) G_{\Xi^S_{k,k+1}(.)}[h]$. Then:

$$G_{\Xi_{k+1|k}}[h] = G_{\Xi^B_{k,k+1}}[h] \int_{\mathcal{F}(\mathcal{X})} (\Phi[h](.))^X p_{\Xi_{k|k}}(X|Z_{1:k})\mu(dX)$$

Using the definition of the PGFl (1.29) again gives:

$$G_{\Xi_{k+1|k}}[h] = G_{\Xi^B_{k,k+1}}[h] G_{\Xi_{k|k}}[\Phi[h]]$$

Then, using the derivation property of the PGFl (1.44) yields, for any $x \in \mathcal{X}$:

$$v_{\Xi_{k+1|k}}(x|Z_{1:k}) K_{\mathcal{X}} = \left[ \frac{\delta G_{\Xi_{k+1|k}}}{\delta x}[h] \right]_{h=1}$$
$$= \underbrace{\left[ \frac{\delta G_{\Xi^B_{k,k+1}}}{\delta x}[h] \right]_{h=1}}_{A} \underbrace{G_{\Xi_{k|k}}[\Phi[1]]}_{B} + \underbrace{G_{\Xi^B_{k,k+1}}[1]}_{C} \underbrace{\left[ \frac{\delta G_{\Xi_{k|k}}}{\delta x}[\Phi[h]] \right]_{h=1}}_{D}$$

Using the derivation property of the PGFl (1.44) again gives:

$$
\begin{aligned}
A &= \left[ \frac{\delta G_{\Xi^B_{k,k+1}}}{\delta x}[h] \right]_{h=1} \\
&= v_{\Xi^B_{k+1|k}}(x) K_{\mathcal{X}}
\end{aligned}
$$

One can note that, for all $x \in \mathcal{X}$:

$$\Phi[1](x) = (1 - p^s_{k,k+1}(x) + p^s_{k,k+1}(x) f^t_{k,k+1}[1|x]) G_{\Xi^S_{k,k+1}(x)}[1]$$

Using the definition of the PGFl (1.29) again gives:

$$
\begin{aligned}
&\Phi[1](x) \\
&= \left( 1 - p^s_{k,k+1}(x) + p^s_{k,k+1}(x) \underbrace{\int_{\mathcal{X}} 1(y) f^t_{k,k+1}(y|x) dy}_{=1} \right) \left( \underbrace{\int_{\mathcal{F}(\mathcal{X})} 1^Y p^S_{k,k+1}(Y|x) \mu(dY)}_{=1} \right) \\
&= 1 - p^s_{k,k+1}(x) + p^s_{k,k+1}(x) \\
&= 1
\end{aligned}
$$

Then, since $\Phi[1] = 1$:

$$
\begin{aligned}
B &= G_{\Xi_{k|k}}[\Phi[1]] \\
&= G_{\Xi_{k|k}}[1]
\end{aligned}
$$

Using the definition of the PGFl (1.29) again gives:

$$
\begin{aligned}
B &= \int_{\mathcal{F}(\mathcal{X})} 1^X p_{\Xi_{k|k}}(X|Z_{1:k}) \mu(dX) \\
&= 1
\end{aligned}
$$

Using the definition of the PGFl (1.29) once more gives:

$$
\begin{aligned}
C &= G_{\Xi^B_{k,k+1}}[1] \\
&= \int_{\mathcal{F}(\mathcal{X})} 1^X p^B_{k,k+1}(X) \mu(dX) \\
&= 1
\end{aligned}
$$

Next, using the definition of the PGFl (1.29) on $D$ yields:

$$D = \left[\frac{\delta G_{\Xi_{k|k}}}{\delta x}[\Phi[h]]\right]_{h=1}$$
$$= v_{\Xi_\Phi}(x)K_{\mathcal{X}}$$

where $\Xi_\Phi$ denotes the RFS with the PGFl $G_{\Xi_{k|k}}[\Phi[.]]$. Denote by $\Xi_{\Phi,y}$ the RFS with PGFl $G_{\Xi_{\Phi,y}}[.] = \Phi[.](y)$. Then, since $\Phi[1] = 1$, property 1.7 applies and thus:

$$D = \int_{\mathcal{X}} v_{\Xi_{\Phi,y}}(x) v_{\Xi_{k|k}}(y|Z_{1:k})dy \; K_{\mathcal{X}}$$
$$= \int_{\mathcal{X}} (v_{\Xi_{\Phi,y}}(x)K_{\mathcal{X}}) v_{\Xi_{k|k}}(y|Z_{1:k})dy$$

Using the definition of the PGFl (1.29) again gives:

$$D = \int_{\mathcal{X}} \left[\frac{\delta G_{\Xi_{\Phi,y}}}{\delta x}[h]\right]_{h=1} v_{\Xi_{k|k}}(y|Z_{1:k})dy$$

But:

$$\frac{\delta G_{\Xi_{\Phi,y}}}{\delta x}[h] = \frac{\delta \Phi[h](y)}{\delta x}$$
$$= \frac{\delta}{\delta x}\left[(1 - p^s_{k,k+1}(y) + p^s_{k,k+1}(y) f^t_{k,k+1}[h|y]) G_{\Xi^S_{k,k+1}(y)}[h]\right]$$
$$= p^s_{k,k+1}(y)\left(\frac{\delta}{\delta x} f^t_{k,k+1}[h|y]\right) G_{\Xi^S_{k,k+1}(y)}[h]$$
$$+ (1 - p^s_{k,k+1}(y) + p^s_{k,k+1}(y) f^t_{k,k+1}[h|y])\left(\frac{\delta}{\delta x} G_{\Xi^S_{k,k+1}(y)}[h]\right)$$

Since $f^t_{k,k+1}[h|y] = \int_{\mathcal{X}} h(z) f^t_{k,k+1}(z|y)dz$, using calculus property (1.48) gives:

$$\frac{\delta G_{\Xi_{\Phi,x}}}{\delta x}[h] = p^s_{k,k+1}(y)\left(f^t_{k,k+1}(x|y)K_{\mathcal{X}}\right) G_{\Xi^S_{k,k+1}(y)}[h]$$
$$+ (1 - p^s_{k,k+1}(y) + p^s_{k,k+1}(y) f^t_{k,k+1}[h|y])\left(\frac{\delta}{\delta x} G_{\Xi^S_{k,k+1}(y)}[h]\right)$$

Therefore, by setting $h = 1$:

$$\left[\frac{\delta G_{\Xi_{\Phi,y}}}{\delta x}[h]\right]_{h=1} = p^s_{k,k+1}(y) f^t_{k,k+1}(x|y) \underbrace{G_{\Xi^S_{k,k+1}(y)}[1]}_{=1} K_{\mathcal{X}}$$
$$+ (1 - p^s_{k,k+1}(y) + p^s_{k,k+1}(y) \underbrace{f^t_{k,k+1}[1|y]}_{=1})\left[\frac{\delta}{\delta x} G_{\Xi^S_{k,k+1}(y)}[h]\right]_{h=1}$$

That is, using the definition of the PGFl (1.29):

$$\frac{\delta G_{\Xi_{\Phi,y}}}{\delta x}[h] = p^s_{k,k+1}(y) f^t_{k,k+1}(x|y) K_{\mathcal{X}} + (1 - p^s_{k,k+1}(y) + p^s_{k,k+1}(y)) v_{\Xi^S_{k,k+1}(y)}(x) K_{\mathcal{X}}$$
$$= p^s_{k,k+1}(y) f^t_{k,k+1}(x|y) K_{\mathcal{X}} + v_{\Xi^S_{k,k+1}(y)}(x) K_{\mathcal{X}}$$

Hence:

$$D = \int_{\mathcal{X}} \left( p^s_{k,k+1}(y) f^t_{k,k+1}(x|y) + v_{\Xi^S_{k,k+1}(y)}(x) \right) v_{\Xi_{k|k}}(y|Z_{1:k}) dy \; K_{\mathcal{X}}$$

Thus:

$$v_{\Xi_{k+1|k}}(x|Z_{1:k}) K_{\mathcal{X}} = AB + CD$$
$$= v_{\Xi^B_{k+1|k}}(x|Z_{1:k}) K_{\mathcal{X}} + \int_{\mathcal{X}} \left( p^s_{k,k+1}(y) f^t_{k,k+1}(x|y) + v_{\Xi^S_{k,k+1}(y)}(x) \right) v_{\Xi_{k|k}}(y|Z_{1:k}) dy \; K_{\mathcal{X}}$$

Finally:

$$v_{\Xi_{k+1|k}}(x|Z_{1:k}) = v_{\Xi^B_{k+1|k}}(x) + \int_{\mathcal{X}} \left( p^s_{k,k+1}(y) f^t_{k,k+1}(x|y) + v_{\Xi^S_{k,k+1}(y)}(x) \right) v_{\Xi_{k|k}}(y|Z_{1:k}) dy$$

□

## Proposition 2.2

This proof is drawn from Mahler's construction of the single-sensor PHD filter in [Mahl 03a].

*Proof.* Let $\Sigma^D_{k+1}(X)$ be the Multi-Bernoulli RFS $\bigcup_{x \in X} \Sigma^D_{k+1}(x)$. Then, using proposition 2.4 on Multi-Bernoulli RFS gives:

$$G_{\Sigma^D_{k+1}(X)}[g] = \prod_{x \in X} \left(1 - p^d_{k+1}(x) + p^d_{k+1}(x) f^o_{k+1}[g|x]\right)$$
$$= (1 - p^d_{k+1}(.) + p^d_{k+1}(.) f^o_{k+1}[g|.])^X$$

Let $\Sigma^C_{k+1}$ be the false alarm RFS. Since is its assumed Poisson with parameter $\lambda^c_{k+1}$ and intensity $\lambda^c_{k+1} c_{k+1}(.)$, using equation (2.3) yields:

$$G_{\Sigma^C_{k+1}}[g] = e^{\lambda^c_{k+1} c_{k+1}[g] - \lambda^c_{k+1}}$$

By construction, the observation RFS $\Sigma^0_{k+1}(X)$ is the union RFS of independent RFSs $\Sigma^D_{k+1}(X)$ and $\Sigma^C_{k+1}$. Thus, using property 1.3 gives:

$$G_{\Sigma^O_{k+1}(X)}[g] = G_{\Sigma^D_{k+1}(X)}[g] G_{\Sigma^C_{k+1}}[g]$$
$$= (1 - p^d_{k+1}(.) + p^d_{k+1}(.) f^o_{k+1}[g|.])^X e^{\lambda^c_{k+1} c_{k+1}[g] - \lambda^c_{k+1}}$$

□

## Proposition 2.3

*Proof.* Let $x \in \mathcal{X}$ be any target state and $z \in \mathcal{Z}$ any measurement state.

$$\beta[g, \delta_x] = \frac{\delta}{\delta x}\beta[g, h]$$
$$= \frac{\delta}{\delta x}\left(\lambda^c_{k+1}c_{k+1}[g] - \lambda^c_{k+1} + v_{\Xi_{k+1|k}}\left[h(1 - p^d_{k+1} + p^d_{k+1}f^o_{k+1}[g|.])\right] - v_{\Xi_{k+1|k}}[1]\right)$$
$$= \frac{\delta}{\delta x}\left(\int_{\mathcal{X}} h(y)(1 - p^d_{k+1}(y) + p^d_{k+1}(y)f^o_{k+1}[g|y])v_{\Xi_{k+1|k}}(y|Z_{1:k})dy\right)$$

Using property (1.48) gives:

$$\beta[g, \delta_x] = (1 - p^d_{k+1}(x) + p^d_{k+1}(x)f^o_{k+1}[g|x])v_{\Xi_{k+1|k}}(x|Z_{1:k})K_{\mathcal{X}}$$

Likewise:

$$\beta[\delta_z, h] = \frac{\delta}{\delta z}\beta[g, h]$$
$$= \frac{\delta}{\delta z}\left(\lambda^c_{k+1}c_{k+1}[g] - \lambda^c_{k+1} + v_{\Xi_{k+1|k}}\left[h(1 - p^d_{k+1})\right] + v_{\Xi_{k+1|k}}\left[p^d_{k+1}f^o_{k+1}[g|.]\right] - v_{\Xi_{k+1|k}}[1]\right)$$
$$= \lambda^c_{k+1}\frac{\delta}{\delta z}\left(\int_{\mathcal{Z}} g(u)c_{k+1}(u)du\right)$$
$$+ \int_{\mathcal{X}} h(y)p^d_{k+1}(y)\frac{\delta}{\delta z}\left(\int_{\mathcal{Z}} g(u)f^o_{k+1}(u|y)du\right)v_{\Xi_{k+1|k}}(y|Z_{1:k})dy$$

Using property (1.48) gives:

$$\beta[\delta_z, h] = \lambda^c_{k+1}c_{k+1}(z)K_{\mathcal{Z}} + \int_{\mathcal{X}} h(y)p^d_{k+1}(y)\underbrace{f^o_{k+1}(z|y)}_{=L^z_{k+1}(y)}v_{\Xi_{k+1|k}}(y|Z_{1:k})dy\ K_{\mathcal{Z}}$$
$$= \lambda^c_{k+1}c_{k+1}(z)K_{\mathcal{Z}} + v_{\Xi_{k+1|k}}[hp^d_{k+1}L^z_{k+1}]K_{\mathcal{Z}}$$

Finally:

$$\beta[\delta_z, \delta_x] = \frac{\delta}{\delta x}\beta[\delta_z, h]$$
$$= \frac{\delta}{\delta x}\left(\lambda^c_{k+1}c_{k+1}(z)K_{\mathcal{Z}} + v_{\Xi_{k+1|k}}[hp^d_{k+1}L^z_{k+1}]K_{\mathcal{Z}}\right)$$
$$= \frac{\delta}{\delta x}\left(\int_{\mathcal{X}} h(y)p^d_{k+1}(y)L^z_{k+1}(y)v_{\Xi_{k+1|k}}(y|Z_{1:k})dy\right)K_{\mathcal{Z}}$$

Using property (1.48) gives:

$$\beta[\delta_z, \delta_x] = p^d_{k+1}(x)L^z_{k+1}(x)v_{\Xi_{k+1|k}}(x|Z_{1:k})K_{\mathcal{X}}K_{\mathcal{Z}}$$

□

## Theorem 2.2

This proof is adapted from one of Mahler's key theorems in the construction of the single-sensor PHD filter (see [Mahl 03a]).

*Proof.* Let $Z_{k+1}$ be the is the set $m_{k+1}$ current measurements, $h$ (resp. $g$) be a real-valued function defined on $\mathcal{X}$ (resp. $\mathcal{Z}$) in [0 1], and $x_0$ any point in $\mathcal{X}$. Then, let $F[g,h]$ be the joint PGFl of PGFls
$G_{\Sigma_{k+1}(X)}[g]$ and $G_{\Xi_{k+1|k}}[h]$. Then:

$$F[g,h] = \int_{\mathcal{F}(\mathcal{Z})} \int_{\mathcal{F}(\mathcal{X})} h^X p_{\Sigma_{k+1}(X)}(Z) p_{\Xi_{k+1|k}}(X|Z_{1:k}) \mu(dX) \mu(dZ)$$
$$= \int_{\mathcal{F}(\mathcal{X})} h^X g^Z \left( \int_{\mathcal{F}(\mathcal{Z})} g^Z p_{\Sigma_{k+1}(X)}(Z) \mu(dZ) \right) p_{\Xi_{k+1|k}}(X|Z_{1:k}) \mu(dX)$$

Which gives, using the definition of the PGFl (1.29):

$$F[g,h] = \int_{\mathcal{F}(\mathcal{X})} h^X G_{\Sigma_{k+1}(X)}[g] p_{\Xi_{k+1|k}}(X|Z_{1:k}) \mu(dX) \tag{30}$$

On the first hand, proposition 2.2 yields:

$$F[g,h] = \int_{\mathcal{F}(\mathcal{X})} h^X (1 - p^d_{k+1}(.) + p^d_{k+1}(.) f^o_{k+1}[g|.])^X e^{\lambda^c_{k+1} c_{k+1}[g] - \lambda^c_{k+1}} p_{\Xi_{k+1|k}}(X|Z_{1:k}) \mu(dX)$$
$$= e^{\lambda^c_{k+1} c_{k+1}[g] - \lambda^c_{k+1}} \int_{\mathcal{F}(\mathcal{X})} \left( h(1 - p^d_{k+1}(.) + p^d_{k+1}(.) f^o_{k+1}[g|.]) \right)^X p_{\Xi_{k+1|k}}(X|Z_{1:k}) \mu(dX)$$
$$= e^{\lambda^c_{k+1} c_{k+1}[g] - \lambda^c_{k+1}} G_{\Xi_{k+1|k}}[h(1 - p^d_{k+1}(.) + p^d_{k+1}(.) f^o_{k+1}[g|.])]$$

Since $\Xi_{k+1|k}$ is assumed Poisson, using (2.3) further simplifies the expression of $F[g,h]$:

$$F[g,h] = e^{\lambda^c_{k+1} c_{k+1}[g] - \lambda^c_{k+1}} e^{v_{\Xi_{k+1|k}}[h(1-p^d_{k+1} + p^d_{k+1} f^o_{k+1}[g|.])] - v_{\Xi_{k+1|k}}[1]}$$
$$= e^{\lambda^c_{k+1} c_{k+1}[g] - \lambda^c_{k+1} + v_{\Xi_{k+1|k}}[h(1-p^d_{k+1} + p^d_{k+1} f^o_{k+1}[g|.])] - v_{\Xi_{k+1|k}}[1]}$$

That is, using the definition of the cross-term (2.19):

$$F[g,h] = e^{\beta[g,h]} \tag{31}$$

On the other hand, derivating (30) in the current measurement set $Z_{k+1}$ gives:

$$\left[ \frac{\delta}{\delta Z_{k+1}} F[g,h] \right]_{g=0} = \int_{\mathcal{F}(\mathcal{X})} h^X \left[ \frac{\delta}{\delta Z_{k+1}} G_{\Sigma_{k+1}(X)}[g] \right]_{g=0} p_{\Xi_{k+1|k}}(X|Z_{1:k}) \mu(dX)$$

Which gives, using the derivative property of PGFl (1.43):

$$\left[\frac{\delta}{\delta Z_{k+1}}F[g,h]\right]_{g=0} = \int_{\mathcal{F}(\mathcal{X})} h^X p_{\Sigma_{k+1}}(Z_{k+1}|X)p_{\Xi_{k+1|k}}(X|Z_{1:k})\mu(dX) \tag{32}$$

But, the PGFl $G_{\Xi_{k+1|k+1}}$ of the posterior RFS $\Xi_{k+1|k+1}$ is by definition (1.29):

$$G_{\Xi_{k+1|k+1}}[h] = \int_{\mathcal{F}(\mathcal{X})} h^X p_{\Xi_{k+1|k+1}}(X|Z_{1:k+1})\mu(dX)$$

Which gives, according to the data update equation of the RFS filter (1.55):

$$G_{\Xi_{k+1|k+1}}[h] = \int_{\mathcal{F}(\mathcal{X})} h^X \left(\frac{p_{\Sigma_{k+1}}(Z_{k+1}|X)p_{\Xi_{k+1|k}}(X|Z_{1:k})}{\int_{\mathcal{F}(\mathcal{X})} p_{\Sigma_{k+1}}(Z_{k+1}|Y)p_{\Xi_{k+1|k}}(Y|Z_{1:k}))\mu(dY)}\right)\mu(dX)$$

Using previous result (32) then (31) yields:

$$G_{\Xi_{k+1|k+1}}[h] = \frac{\left[\frac{\delta}{\delta Z_{k+1}}e^{\beta[g,h]}\right]_{g=0}}{\left[\frac{\delta}{\delta Z_{k+1}}e^{\beta[g,h]}\right]_{g=0,h=1}}$$

Thus, using the derivative property of PGFl (1.44) gives:

$$v_{\Xi_{k+1|k+1}}(x_0) = \frac{\left[\frac{\delta}{\delta x_0}\left(\frac{\delta}{\delta Z_{k+1}}e^{\beta[g,h]}\right)\right]_{g=0,h=1}}{\left[\frac{\delta}{\delta Z_{k+1}}e^{\beta[g,h]}\right]_{g=0,h=1}}K_{\mathcal{X}}^{-1} \tag{33}$$

Thus, the posterior PHD $v_{\Xi_{k+1|k+1}}(.)$ can be computed *exclusively* with derivatives of the cross-term $\beta[g,h]$. Equation (33) can be further simplified as follows:

$$v_{\Xi_{k+1|k+1}}(x_0) = \frac{\left[\frac{\delta}{\delta x_0}\left(\frac{\delta}{\delta Z_{k+1}}e^{\beta[g,h]}\right)\right]_{g=0,h=1}}{\left[\frac{\delta}{\delta Z_{k+1}}e^{\beta[g,h]}\right]_{g=0,h=1}}K_{\mathcal{X}}^{-1}$$

Recall that cross-terms vanish when derivated in more than one measurement. Thus:

$$
\begin{aligned}
v_{\Xi_{k+1|k+1}}(x_0) &= \frac{\left[\frac{\delta}{\delta x_0}\left(e^{\beta[g,h]}\prod_{i=1}^{m_{k+1}}\beta[\delta_{z_{i,k+1}},h]\right)\right]_{g=0,h=1}}{\left[e^{\beta[g,h]}\prod_{i=1}^{m_{k+1}}\beta[\delta_{z_{i,k+1}},h]\right]_{g=0,h=1}}K_{\mathcal{X}}^{-1} \\
&= \frac{\left[\beta[g,\delta_x]e^{\beta[g,h]}\prod_{i=1}^{m_{k+1}}\beta[\delta_{z_{i,k+1}},h]\right]_{g=0,h=1}}{\left[e^{\beta[g,h]}\prod_{i=1}^{m_{k+1}}\beta[\delta_{z_{i,k+1}},h]\right]_{g=0,h=1}}K_{\mathcal{X}}^{-1} \\
&\quad + \frac{\left[e^{\beta[g,h]}\sum_{i=1}^{m_{k+1}}\left(\beta[\delta_{z_{i,k+1}},\delta_x]\prod_{j\neq i}\beta[\delta_{z_{i,k+1}},h]\right)\right]_{g=0,h=1}}{\left[e^{\beta[g,h]}\prod_{i=1}^{m_{k+1}}\beta[\delta_{z_{i,k+1}},h]\right]_{g=0,h=1}}K_{\mathcal{X}}^{-1} \\
&= \left(\beta[\delta_\emptyset,\delta_x] + \sum_{z\in Z_{k+1}}\frac{\beta[\delta_z,\delta_x]}{\beta[\delta_z,1]}\right)K_{\mathcal{X}}^{-1}
\end{aligned}
\tag{34}
$$

That is, using the definition of the cross-term (2.19):

$$
\begin{aligned}
&v_{\Xi_{k+1|k+1}}(x_0) \\
&= \left((1-p_{k+1}^d(x))v_{\Xi_{k+1|k}}(x|Z_{1:k})K_{\mathcal{X}}\right. \\
&\quad \left. + \sum_{z\in Z_{k+1}}\frac{p_{k+1}^d(x)L_{k+1}^z(x)v_{\Xi_{k+1|k}}(x|Z_{1:k})K_{\mathcal{X}}K_{\mathcal{Z}}}{\lambda_{k+1}^c c_{k+1}(z)K_{\mathcal{Z}} + v_{\Xi_{k+1|k}}[p_{k+1}^d L_{k+1}^z]K_{\mathcal{Z}}}\right)K_{\mathcal{X}}^{-1} \\
&= \left(1-p_{k+1}^d(x) + \sum_{z\in Z_{k+1}}\frac{p_{k+1}^d(x)L_{k+1}^z(x)}{\lambda_{k+1}^c c_{k+1}(z) + v_{\Xi_{k+1|k}}[p_{k+1}^d L_{k+1}^z]}\right)v_{\Xi_{k+1|k}}(x|Z_{1:k})
\end{aligned}
\tag{35}
$$

□

## Proposition 2.5

*Proof.* Let $x_0 \in \mathcal{X}$ be any target state, $\{z^j\}_{j=1}^S$ be any measurement family where $z^j \in \mathcal{Z}^j$, and $Z$ any subset of $\{z^j\}_{j=1}^S$. First:

$$\begin{aligned}
&\beta[\delta_\emptyset, \bar{g}, \delta_{x_0}] = \frac{\delta}{\delta x_0}\beta[g^1, ..., g^S, h] \\
&= \frac{\delta}{\delta x_0}\left(\sum_{j=1}^{S}(\lambda_{k+1}^{c,j}c_{k+1}^{j}[g^j] - \lambda_{k+1}^{c,j}) \right. \\
&\qquad \left. + v_{\Xi_{k+1|k}}\left[h\left(\prod_{j=1}^{S}(1 - p_{k+1}^{d,j} + p_{k+1}^{d,j}f_{k+1}^{o,j}[g^j|.])\right)\right] - v_{\Xi_{k+1|k}}[1]\right) \\
&= \frac{\delta}{\delta x_0}\left(\int_{\mathcal{X}} h(x)\prod_{j=1}^{S}(1 - p_{k+1}^{d,j}(x) + p_{k+1}^{d,j}(x)f_{k+1}^{o,j}[g^j|x])v_{\Xi_{k+1|k}}(x|Z_{1:k})dx\right)
\end{aligned}$$

Using calculus property (1.48) gives:

$$\beta[\delta_\emptyset, \bar{g}, \delta_{x_0}] = \prod_{j=1}^{S}(1 - p_{k+1}^{d,j}(x_0) + p_{k+1}^{d,j}(x_0)f_{k+1}^{o,j}[g^j|x])v_{\Xi_{k+1|k}}(x_0|Z_{1:k})K_{\mathcal{X}}$$

Then:

$$\begin{aligned}
&\beta[\delta_Z, \bar{g}, h] = \frac{\delta}{\delta Z}\beta[g^1, ..., g^S, h] \\
&= \frac{\delta}{\delta Z}\left(\sum_{j=1}^{S}(\lambda_{k+1}^{c,j}c_{k+1}^{j}[g^j] - \lambda_{k+1}^{c,j}) \right. \\
&\qquad \left. + v_{\Xi_{k+1|k}}\left[h\left(\prod_{j=1}^{S}(1 - p_{k+1}^{d,j} + p_{k+1}^{d,j}f_{k+1}^{o,j}[g^j|.])\right)\right] - v_{\Xi_{k+1|k}}[1]\right) \\
&= \begin{cases}
\dfrac{\delta}{\delta z^{j_0}}\left(\displaystyle\int_{\mathcal{Z}^{j_0}} \lambda_{k+1}^{c,j_0}c_{k+1}^{j_0}(z)dz\right) + \displaystyle\int_{\mathcal{X}} h(x)\frac{\delta}{\delta z^{j_0}}\left(\int_{\mathcal{Z}^{j_0}} p_{k+1}^{d,j_0}(x)f_{k+1}^{o,j_0}[g^{j_0}|x])dz\right) \\
\qquad \times \displaystyle\prod_{z^j \neq z^{j_0}}(1 - p_{k+1}^{d,j}(x) + p_{k+1}^{d,j}(x)f_{k+1}^{o,j}[g^j|x])v_{\Xi_{k+1|k}}(x|Z_{1:k})dx \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (Z = \{z^{j_0}\}) \\
\displaystyle\int_{\mathcal{X}} h(x)\prod_{z^j \in Z}\left(\frac{\delta}{\delta z^j}\left(\int_{\mathcal{Z}^j} p_{k+1}^{d,j}(x)f_{k+1}^{o,j}[g^j|x])dz\right)\right) \\
\qquad \times \displaystyle\prod_{z^j \notin Z}(1 - p_{k+1}^{d,j}(x) + p_{k+1}^{d,j}(x)f_{k+1}^{o,j}[g^j|x])v_{\Xi_{k+1|k}}(x|Z_{1:k})dx \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (|Z| \geqslant 2)
\end{cases}
\end{aligned}$$

Using calculus property (1.48) gives:

$$
\begin{aligned}
&\beta[\delta_Z, \bar{g}, h] \\
&= \begin{cases}
\lambda_{k+1}^{c,j_0} c_{k+1}^{j_0}(z^{j_0}) K_{\mathcal{Z}^{j_0}} + \displaystyle\int_{\mathcal{X}} h(x) p_{k+1}^{d,j_0}(x) \underbrace{f_{k+1}^{o,j_0}(z^{j_0}|x)}_{=L_{k+1}^{z^{j_0},j_0}(x)} K_{\mathcal{Z}^{j_0}} \\
\qquad \times \displaystyle\prod_{z^j \neq z^{j_0}} (1 - p_{k+1}^{d,j}(x) + p_{k+1}^{d,j}(x) f_{k+1}^{o,j}[g^j|x]) v_{\Xi_{k+1|k}}(x|Z_{1:k}) dx \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (Z = \{z^{j_0}\}) \\
\displaystyle\int_{\mathcal{X}} h(x) \prod_{z^j \in Z} (p_{k+1}^{d,j}(x) \underbrace{f_{k+1}^{o,j}(z^j|x)}_{=L_{k+1}^{z^j,j}(x)} K_{\mathcal{Z}^j}) \\
\qquad \times \displaystyle\prod_{z^j \notin Z} (1 - p_{k+1}^{d,j}(x) + p_{k+1}^{d,j}(x) f_{k+1}^{o,j}[g^j|x]) v_{\Xi_{k+1|k}}(x|Z_{1:k}) dx \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (|Z| \geqslant 2)
\end{cases}
\end{aligned}
$$

$$
\begin{aligned}
&= \begin{cases}
\lambda_{k+1}^{c,j_0} c_{k+1}^{j_0}(z^{j_0}) K_{\mathcal{Z}^{j_0}} \\
\qquad + v_{\Xi_{k+1|k}}[h p_{k+1}^{d,j_0} L_{k+1}^{z^{j_0},j_0} K_{\mathcal{Z}^{j_0}} \displaystyle\prod_{z^j \neq z^{j_0}} (1 - p_{k+1}^{d,j} + p_{k+1}^{d,j} f_{k+1}^{o,j}[g^j|.])] \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (Z = \{z^{j_0}\}) \\
v_{\Xi_{k+1|k}}[h \displaystyle\prod_{z^j \in Z} (p_{k+1}^{d,j} L_{k+1}^{z^j,j} K_{\mathcal{Z}^j}) \prod_{z^j \notin Z} (1 - p_{k+1}^{d,j} + p_{k+1}^{d,j} f_{k+1}^{o,j}[g^j|.])] \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (|Z| \geqslant 2)
\end{cases}
\end{aligned}
$$

Finally:

$$\beta[\delta_Z, \bar{g}, \delta_{x_0}] = \frac{\delta}{\delta x_0}\beta[\delta_Z, \bar{g}, h]$$

$$= \begin{cases} \dfrac{\delta}{\delta x_0}\left( \displaystyle\int_{\mathcal{X}} h(x) p_{k+1}^{d,j_0}(x) L_{k+1}^{z^{j_0},j_0}(x) K_{\mathcal{Z}^{j_0}} \right. \\ \qquad \left. \times \displaystyle\prod_{z^j \neq z^{j_0}} (1 - p_{k+1}^{d,j}(x) + p_{k+1}^{d,j}(x) f_{k+1}^{o,j}[g^j|x]) v_{\Xi_{k+1|k}}(x|Z_{1:k}) dx \right) \\ \qquad\qquad (Z = \{z^{j_0}\}) \\ \dfrac{\delta}{\delta x_0}\left( \displaystyle\int_{\mathcal{X}} h(x) \prod_{z^j \in Z} (p_{k+1}^{d,j}(x) L_{k+1}^{z^j,j}(x) K_{\mathcal{Z}^j}) \right. \\ \qquad \left. \times \displaystyle\prod_{z^j \notin Z} (1 - p_{k+1}^{d,j}(x) + p_{k+1}^{d,j}(x) f_{k+1}^{o,j}[g^j|x]) v_{\Xi_{k+1|k}}(x|Z_{1:k}) dx \right) \\ \qquad\qquad (|Z| \geqslant 2) \end{cases}$$

$$= \frac{\delta}{\delta x_0}\left( \int_{\mathcal{X}} h(x) \prod_{z^j \in Z} (p_{k+1}^{d,j}(x) L_{k+1}^{z^j,j}(x) K_{\mathcal{Z}^j}) \right.$$
$$\left. \times \prod_{z^j \notin Z} (1 - p_{k+1}^{d,j}(x) + p_{k+1}^{d,j}(x) f_{k+1}^{o,j}[g^j|x]) v_{\Xi_{k+1|k}}(x|Z_{1:k}) dx \right)$$

Using calculus property (1.48) gives:

$$\beta[\delta_Z, \bar{g}, \delta_{x_0}] = \prod_{z^j \in Z} (p_{k+1}^{d,j}(x_0) L_{k+1}^{z^j,j}(x_0) K_{\mathcal{Z}^j})$$
$$\times \prod_{z^j \notin Z} (1 - p_{k+1}^{d,j}(x_0) + p_{k+1}^{d,j}(x_0) f_{k+1}^{o,j}[g^j|x_0]) v_{\Xi_{k+1|k}}(x_0|Z_{1:k}) K_{\mathcal{X}}$$

□

## Theorem 2.3

This proof is an extension of Mahler's proof in the single-sensor case (see [Mahl 03a]). Note that a construction of the two-sensor case is also provided by Mahler in [Mahl 09a].

*Proof.* Let $Z_{k+1} = \bigsqcup_{j=1}^{S} Z_{k+1}^j$ be the set of $m_{k+1} = \sum_{j=1}^{S} m_{k+1}^j$ current measurements, $h$ (resp. $g^j$) be a real-valued function defined on $\mathcal{X}$ (resp. $\mathcal{Z}^j$) in [0 1], and $x_0$

any point in $\mathcal{X}$. Then, let $F[g^1,...g^S,h]$ be the joint PGFl of PGFls $G_{\Sigma^1_{k+1}(X)}[g]$,..., $G_{\Sigma^S_{k+1}(X)}[g]$ and $G_{\Xi_{k+1|k}}[h]$. Then:

$$\begin{aligned}
&F[g^1,...g^S,h]\\
&=\int_{\mathcal{F}(\mathcal{Z}^1)}...\int_{\mathcal{F}(\mathcal{Z}^S)}\int_{\mathcal{F}(\mathcal{X})}h^X\prod_{j=1}^{S}\left(g^{Z^j}p_{\Sigma^j_{k+1}(X)}(Z^j)\right)p_{\Xi_{k+1|k}}(X|Z_{1:k})\mu(dX)\mu(dZ^1)...\mu(dZ^S)\\
&=\int_{\mathcal{F}(\mathcal{X})}h^X\prod_{j=1}^{S}\left(\int_{\mathcal{F}(\mathcal{Z}^j)}g^Z p_{\Sigma^j_{k+1}(X)}(Z)\mu(dZ)\right)p_{\Xi_{k+1|k}}(X|Z_{1:k})\mu(dX)
\end{aligned}$$

Which gives, using the definition of the PGFl (1.29):

$$F[g^1,...g^S,h]=\int_{\mathcal{F}(\mathcal{X})}h^X\prod_{j=1}^{S}(G_{\Sigma^j_{k+1}(X)}[g^j])p_{\Xi_{k+1|k}}(X|Z_{1:k})\mu(dX) \tag{36}$$

On the first hand, proposition 2.4 yields:

$$\begin{aligned}
&F[g^1,...g^S,h]\\
&=\int_{\mathcal{F}(\mathcal{X})}h^X\prod_{j=1}^{S}\left((1-p^{d,j}_{k+1}(.)+p^{d,j}_{k+1}(.)f^{o,j}_{k+1}[g^j|.])^X e^{\lambda^{c,j}_{k+1}c^j_{k+1}[g^j]-\lambda^{c,j}_{k+1}}\right)p_{\Xi_{k+1|k}}(X|Z_{1:k})\mu(dX)\\
&=\prod_{j=1}^{S}\left(e^{\lambda^{c,j}_{k+1}c^j_{k+1}[g^j]-\lambda^{c,j}_{k+1}}\right)\\
&\quad\times\int_{\mathcal{F}(\mathcal{X})}\left(h\prod_{j=1}^{S}\left(1-p^{d,j}_{k+1}(.)+p^{d,j}_{k+1}(.)f^{o,j}_{k+1}[g^j|.]\right)\right)^X p_{\Xi_{k+1|k}}(X|Z_{1:k})\mu(dX)\\
&=\prod_{j=1}^{S}\left(e^{\lambda^{c,j}_{k+1}c^j_{k+1}[g^j]-\lambda^{c,j}_{k+1}}\right)G_{\Xi_{k+1|k}}\left[h\prod_{j=1}^{S}\left(1-p^{d,j}_{k+1}(.)+p^{d,j}_{k+1}(.)f^{o,j}_{k+1}[g^j|.]\right)\right]
\end{aligned}$$

Since $\Xi_{k+1|k}$ is assumed Poisson, using (2.3) further simplifies the expression of $F[g^1,...g^S,h]$:

$$\begin{aligned}
&F[g^1,...g^S,h]=\prod_{j=1}^{S}\left(e^{\lambda^{c,j}_{k+1}c^j_{k+1}[g^j]-\lambda^{c,j}_{k+1}}\right)e^{v_{\Xi_{k+1|k}}[h\prod_{j=1}^{S}\left(1-p^{d,j}_{k+1}(.)+p^{d,j}_{k+1}(.)f^{o,j}_{k+1}[g^j|.]\right)]-v_{\Xi_{k+1|k}}[1]}\\
&=e^{\sum_{j=1}^{S}\left(\lambda^{c,j}_{k+1}c^j_{k+1}[g^j]-\lambda^{c,j}_{k+1}\right)+v_{\Xi_{k+1|k}}[h\prod_{j=1}^{S}\left(1-p^{d,j}_{k+1}(.)+p^{d,j}_{k+1}(.)f^{o,j}_{k+1}[g^j|.]\right)]-v_{\Xi_{k+1|k}}[1]}
\end{aligned}$$

That is, using the definition of the cross-term (2.33):

$$F[g^1,...g^S,h]=e^{\beta[g^1,...g^S,h]} \tag{37}$$

On the other hand, derivating (36) in the current measurement set $Z_{k+1}$ gives:

$$\left[\frac{\delta}{\delta Z_{k+1}}F[g^1,...g^S,h]\right]_{g^{1...S}=0}$$
$$=\int_{\mathcal{F}(\mathcal{X})}h^X\prod_{j=1}^{S}\left[\frac{\delta}{\delta Z_{k+1}^j}G_{\Sigma_{k+1}^j(X)}[g^j]\right]_{g^j=0}p_{\Xi_{k+1|k}}(X|Z_{1:k})\mu(dX)$$

Which gives, using the derivative property of PGFl (1.43):

$$\left[\frac{\delta}{\delta Z_{k+1}}F[g^1,...g^S,h]\right]_{g^{1...S}=0}=\int_{\mathcal{F}(\mathcal{X})}h^X\prod_{j=1}^{S}(p_{\Sigma_{k+1}^j}(Z_{k+1}^j|X))p_{\Xi_{k+1|k}}(X|Z_{1:k})\mu(dX)$$

That is, using the independence of the single-sensor observation processes (2.32):

$$\left[\frac{\delta}{\delta Z_{k+1}}F[g^1,...g^S,h]\right]_{g^{1...S}=0}=\int_{\mathcal{F}(\mathcal{X})}h^X p_{\Sigma_{k+1}}(Z_{k+1}|X))p_{\Xi_{k+1|k}}(X|Z_{1:k})\mu(dX) \quad (38)$$

But, the PGFl $G_{\Xi_{k+1|k+1}}$ of the posterior RFS $\Xi_{k+1|k+1}$ is by definition (1.29):

$$G_{\Xi_{k+1|k+1}}[h]=\int_{\mathcal{F}(\mathcal{X})}h^X p_{\Xi_{k+1|k+1}}(X|Z_{1:k+1})\mu(dX)$$

Which gives, according to the data update equation of the RFS filter (1.55):

$$G_{\Xi_{k+1|k+1}}[h]=\int_{\mathcal{F}(\mathcal{X})}h^X\left(\frac{p_{\Sigma_{k+1}}(Z_{k+1}|X)p_{\Xi_{k+1|k}}(X|Z_{1:k})}{\int_{\mathcal{F}(\mathcal{X})}p_{\Sigma_{k+1}}(Z_{k+1}|Y)p_{\Xi_{k+1|k}}(Y|Z_{1:k}))\mu(dY)}\right)\mu(dX)$$

Using previous result (38) then (37) yields:

$$G_{\Xi_{k+1|k+1}}[h]=\frac{\left[\frac{\delta}{\delta Z_{k+1}}e^{\beta[g^1,...,g^S,h]}\right]_{g^{1...S}=0}}{\left[\frac{\delta}{\delta Z_{k+1}}e^{\beta[g^1,...,g^S,h]}\right]_{g^{1...S}=0,h=1}}$$

Thus, using the derivative property of PGFl (1.44) gives:

$$v_{\Xi_{k+1|k+1}}(x_0)=\frac{\left[\frac{\delta}{\delta x_0}\left(\frac{\delta}{\delta Z_{k+1}}e^{\beta[g^1,...,g^S,h]}\right)\right]_{g^{1...S}=0,h=1}}{\left[\frac{\delta}{\delta Z_{k+1}}e^{\beta[g^1,...,g^S,h]}\right]_{g^{1...S}=0,h=1}}K_{\mathcal{X}}^{-1}$$

□

## Lemma 2.1

*Proof.* Let $s$ be any integer $s < S$, $\{Z^j\}_{j=1}^{s+1}$ be any family of finite subsets $Z^j \subset \mathcal{Z}^j$ with $Z^{s+1} = \{z_i^{s+1}\}_{i=1}^{m^{s+1}}$. First, let us show that:

$$\begin{aligned}&\mathcal{C}(Z^1, ..., Z^{s+1})\\ &= \{C \cup \mathcal{M}(Z) \mid Z \subseteq Z^{s+1}, C \in \mathcal{C}(Z^1, ..., Z^s, Z^{s+1} \setminus Z), C \cap \mathcal{M}(Z^{s+1}) = \emptyset\} \quad (39)\end{aligned}$$

That is, any combinational term based on all the measurements in $Z^{s+1}$ can be uniquely decomposed as the union of:

- all the singletons of measurements from some $Z \subseteq Z^{s+1}$ ("$\mathcal{M}(Z)$");
- a combinational term based on the remaining terms in $Z^{s+1}$ ("$C \in \mathcal{C}(Z^1, ..., Z^s, Z^{s+1} \setminus Z), C \cap \mathcal{M}(Z^{s+1}) = \emptyset$").

Let $A \in \mathcal{C}(Z^1, ..., Z^{s+1})$ be any combinational term. Denote by $Z_A \subseteq Z^{s+1}$ the subset $Z_A = \{z \in Z^{s+1} \mid \{z\} \in A\}$. Using the definition of the term set (2.47) gives $\mathcal{M}(Z_A) = \chi(Z_A) = \{\{z\} \mid z \in Z_A\}$ and thus $A = \mathcal{M}(Z_A) \cup C_A$ where $C_A = A \setminus \mathcal{M}(Z_A)$. By construction, $C_A \subseteq \mathcal{M}(Z^1, ..., Z^s, Z^{s+1} \setminus Z_A)$ and $C_A \cap \mathcal{M}(Z^{s+1}) = \emptyset$. Besides, $\varphi_{Z^1,...,Z^{s+1}}(A) = 1$ and $\varphi_Z(\mathcal{M}(Z_A)) = 1$ imply that $\varphi_{Z^1,...,Z^s,Z^{s+1} \setminus Z_A}(C_A) = 1$, that is, $C_A \in \mathcal{C}(Z^1, ..., Z^s, Z^{s+1} \setminus Z_A)$. By construction, the decomposition $Z_A, C_A$ is unique: if $A = C_A \cup \mathcal{M}(Z_A) = C_B \cup \mathcal{M}(Z_B)$, the conditions $C_A \cap \mathcal{M}(Z^{s+1}) = \emptyset$ and $C_B \cap \mathcal{M}(Z^{s+1}) = \emptyset$ imply that $\mathcal{M}(Z_A) = \mathcal{M}(Z_B)$, thus $Z_A = Z_B$ and $C_A = C_B$.

Conversely, let $Z \subseteq Z^{s+1}$ be any subset of $Z^{s+1}$, and $C \in \mathcal{C}(Z^1, ..., Z^s, Z^{s+1} \setminus Z)$ any combinational term such that $C \cap \mathcal{M}(Z^{s+1}) = \emptyset$. Then, $C \subseteq \mathcal{M}(Z^1, ..., Z^s, Z^{s+1} \setminus Z)$ and therefore $(C \cup \mathcal{M}(Z)) \subseteq \mathcal{M}(Z^1, ..., Z^s, Z^{s+1})$. Besides, $\varphi_Z(\mathcal{M}(Z)) = 1$ and $\varphi_{Z^1,...,Z^s,Z^{s+1} \setminus Z}(C) = 1$, thus $\varphi_{Z^1,...,Z^s,Z^{s+1}}(C \cup \mathcal{M}(Z)) = 1$.

Therefore, equality (39) is true. Thus we can write:

$$\begin{aligned}&\mathcal{C}(Z^{1:s+1})\\ &= \{C \cup \mathcal{M}(Z) \mid Z \subseteq Z^{s+1}, C \in \mathcal{C}(Z^{1...s}, Z^{s+1} \setminus Z), C \cap \mathcal{M}(Z^{s+1}) = \emptyset\}\end{aligned}$$

$$= \bigcup_{Z \subseteq Z^{s+1}} \left\{ C \cup \mathcal{M}(Z) \mid C \subseteq \mathcal{M}(Z^{1\ldots s}, Z^{s+1} \setminus Z), \varphi_{Z^{1\ldots s}, Z^{s+1}\setminus Z}(C) = 1 \right\}$$

$$= \bigcup_{n=0}^{m^{s+1}} \bigcup_{\substack{I \subseteq [1\ m^{s+1}] \\ |I| = ns}} \left\{ C^a \cup \left( \bigcup_{i \in I} \left\{ \{ z_i^{s+1} \} \cup C_i^b \right\} \right) \cup \left( \bigcup_{i \notin I} \left\{ \{ z_i^{s+1} \} \right\} \right) \right.$$

$$\left. \mid C^a, C^b \subseteq \mathcal{M}(Z^{1:s}), |C^b| = n, \varphi_{Z^{1:s}}(C^a \cup C^b) = 1 \right\}$$

$$= \bigcup_{n=0}^{m^{s+1}} \bigcup_{\substack{C \subseteq \mathcal{M}(Z^{1:s}) \\ |C| \geqslant n \\ \varphi_{Z^{1:s}}(C)=1}} \bigcup_{\substack{I \subseteq [1\ m^{s+1}] \\ J \subseteq [1\ |C|] \\ |I|=|J|=n}} \bigcup_{\sigma \in Bij(I,J)}$$

$$\left( \bigcup_{j \notin J} \{ C_j^a \} \right) \cup \left( \bigcup_{i \in I} \left\{ \{ z_i^{s+1} \} \cup C_{\sigma(i)}^b \right\} \right) \cup \left( \bigcup_{i \notin I} \left\{ \{ z_i^{s+1} \} \right\} \right)$$

That is, using the definition of combinational terms (2.49):

$$\mathcal{C}(Z^{1:s+1})$$

$$= \bigcup_{n=0}^{m^{s+1}} \bigcup_{\substack{C \in \mathcal{C}(Z^{1:s}) \\ |C| \geqslant n}} \bigcup_{\substack{I \subseteq [1\ m^{s+1}] \\ J \subseteq [1\ |C|] \\ |I|=|J|=n}} \bigcup_{\sigma \in Bij(I,J)} U_{I,J}^{\sigma}(Z^{s+1}, C)$$

$$= \bigcup_{C \in \mathcal{C}(Z^{1:s})} \bigcup_{n=0}^{\min(|C|, m^{s+1})} \bigcup_{\substack{I \subseteq [1\ m^{s+1}] \\ J \subseteq [1\ |C|] \\ |I|=|J|=n}} \bigcup_{\sigma \in Bij(I,J)} U_{I,J}^{\sigma}(Z^{s+1}, C)$$

□

## Theorem 2.4

*Proof.* Let $Z_{k+1} = \bigsqcup_{j=1}^{S} Z_{k+1}^j$ be the set of $m_{k+1} = \sum_{j=1}^{S} m_{k+1}^j$ current measurements, $h$ (resp. $g^j$) be a real-valued function defined on $\mathcal{X}$ (resp. $\mathcal{Z}^j$) in [0 1], and $x_0$ any point in $\mathcal{X}$. First, let us prove by induction on $1 \leqslant s \leqslant S$ that:

$$\frac{\delta^s}{\delta Z_{k+1}^1 \ldots \delta Z_{k+1}^s} e^{\beta[g^1, \ldots, g^S, h]} = e^{\beta[\delta_\emptyset, \bar{g}, h]} \sum_{C \in \mathcal{C}(Z_{k+1}^{1\ldots s})} \prod_{C_i \in C} \beta[\delta_{C_i}, \bar{g}, h] \tag{40}$$

Let us consider the basis case $s = 1$. Recall from proposition 2.5 that a cross-terms vanishes when derivated in two points from the same observation space. Thus:

$$\begin{aligned}\frac{\delta}{\delta Z_{k+1}^1} e^{\beta[\delta_\emptyset, \bar{g}, h]} &= e^{\beta[\delta_\emptyset, \bar{g}, h]} \prod_{z \in Z_{k+1}^1} \beta[\delta_{\{z\}}, \bar{g}, h] \\ &= e^{\beta[\delta_\emptyset, \bar{g}, h]} \prod_{Z \in \mathcal{M}(Z_{k+1}^1)} \beta[\delta_Z, \bar{g}, h] \\ &= e^{\beta[\delta_\emptyset, \bar{g}, h]} \sum_{C \in \mathcal{C}(Z_{k+1}^1)} \prod_{Z \in C} \beta[\delta_Z, \bar{g}, h]\end{aligned}$$

Therefore, the case holds for $s = 1$. Assuming that the case holds for $s$, $s < S$, let us prove that it holds for $s + 1$. We can write:

$$\frac{\delta^{s+1}}{\delta Z_{k+1}^1 ... \delta Z_{k+1}^s \delta Z_{k+1}^{s+1}} e^{\beta[\delta_\emptyset, \bar{g}, h]} = \frac{\delta}{\delta Z_{k+1}^{s+1}} \left( \frac{\delta^s}{\delta Z_{k+1}^1 ... \delta Z_{k+1}^s} e^{\beta[g^1, ..., g^S, h]} \right)$$

Thus, by using the case at step $s$:

$$\frac{\delta^{s+1}}{\delta Z_{k+1}^1 ... \delta Z_{k+1}^s \delta Z_{k+1}^{s+1}} e^{\beta[\delta_\emptyset, \bar{g}, h]} = \frac{\delta}{\delta Z_{k+1}^{s+1}} \left( e^{\beta[\delta_\emptyset, \bar{g}, h]} \sum_{C \in \mathcal{C}(Z_{k+1}^{1...s})} \prod_{Z \in C} \beta[\delta_Z, \bar{g}, h] \right)$$

$$= \sum_{n=0}^{m_{k+1}^{s+1}} \sum_{\substack{I \subseteq [1 \ m_{k+1}^{s+1}] \\ |I| = n}} \left( \frac{\delta}{\delta \{z_{i,k+1}^{s+1}\}_{i \notin I}} e^{\beta[\delta_\emptyset, \bar{g}, h]} \frac{\delta}{\delta \{z_{i,k+1}^{s+1}\}_{i \in I}} \left( \sum_{C \in \mathcal{C}(Z_{k+1}^{1...s})} \prod_{Z \in C} \beta[\delta_Z, \bar{g}, h] \right) \right)$$

$$= \sum_{n=0}^{m_{k+1}^{s+1}} \sum_{\substack{I \subseteq [1 \ m_{k+1}^{s+1}] \\ |I| = n}} \left( e^{\beta[\delta_\emptyset, \bar{g}, h]} \left( \prod_{i \notin I} \beta[\delta_{\{z_{i,k+1}^{s+1}\}}, \bar{g}, h] \right) \sum_{C \in \mathcal{C}(Z_{k+1}^{1...s})} \left( \underbrace{\frac{\delta^n}{\delta \{z_{i,k+1}^{s+1}\}_{i \in I}} \prod_{Z \in C} \beta[\delta_Z, \bar{g}, h]}_{=A} \right) \right)$$

Recall from proposition 2.5 that a cross-term vanishes when derivated in two points from the same observation space. Thus, $A$ can be expanded as follows:

$$A = \begin{cases} 0 & (n > |C|) \\ \displaystyle\sum_{\substack{J \subseteq [1 \ |C|] \\ |J| = n}} \sum_{\sigma \in Bij(I,J)} \prod_{j \notin J} \beta[\delta_{C_j}, \bar{g}, h] \prod_{i \in I} \beta[\delta_{\{z_{i,k+1}^{s+1}\} \cup C_{\sigma(i)}}, \bar{g}, h] & (n \leqslant |C|) \end{cases}$$

And therefore:

$$\frac{\delta^{s+1}}{\delta Z_{k+1}^1 ... \delta Z_{k+1}^s \delta Z_{k+1}^{s+1}} e^{\beta[\delta_\emptyset, \bar{g}, h]}$$
$$= e^{\beta[\delta_\emptyset, \bar{g}, h]} \sum_{n=0}^{m_{k+1}^{s+1}} \sum_{\substack{I \subseteq [1 \ m_{k+1}^{s+1}] \\ |I|=n}} \sum_{\substack{C \in \mathcal{C}(Z_{k+1}^{1...s}) \\ |C| \geqslant n}} \sum_{\substack{J \subseteq [1 \ |C|] \\ |J|=n}} \sum_{\sigma \in Bij(I,J)}$$
$$\left( \prod_{i \notin I} \beta[\delta_{\{z_{i,k+1}^{s+1}\}}, \bar{g}, h] \right) \left( \prod_{j \notin J} \beta[\delta_{C_j}, \bar{g}, h] \right) \left( \prod_{i \in I} \beta[\delta_{\{z_{i,k+1}^{s+1}\} \cup C_{\sigma(i)}}, \bar{g}, h] \right)$$
$$= e^{\beta[\delta_\emptyset, \bar{g}, h]} \sum_{C \in \mathcal{C}(Z_{k+1}^{1...s})} \sum_{n=0}^{\min(|C|, m_{k+1}^{s+1})} \sum_{\substack{I \subseteq [1 \ m_{k+1}^{s+1}] \\ J \subseteq [1 \ |C|] \\ |I|=|J|=n}} \sum_{\sigma \in Bij(I,J)}$$
$$\left( \prod_{i \notin I} \beta[\delta_{\{z_{i,k+1}^{s+1}\}}, \bar{g}, h] \right) \left( \prod_{j \notin J} \beta[\delta_{C_j}, \bar{g}, h] \right) \left( \prod_{i \in I} \beta[\delta_{\{z_{i,k+1}^{s+1}\} \cup C_{\sigma(i)}}, \bar{g}, h] \right)$$

That is, using lemma 2.1:

$$\frac{\delta^{s+1}}{\delta Z_{k+1}^1 ... \delta Z_{k+1}^s \delta Z_{k+1}^{s+1}} e^{\beta[\delta_\emptyset, \bar{g}, h]} = e^{\beta[\delta_\emptyset, \bar{g}, h]} \sum_{C \in \mathcal{C}(Z_{k+1}^{1...s+1})} \prod_{C_i \in C} \beta[\delta_{C_i}, \bar{g}, h]$$

Therefore, the case holds true for each $s \leqslant S$ and for $s = S$ equation (40) becomes:

$$\frac{\delta}{\delta Z_{k+1}} e^{\beta[\delta_\emptyset, \bar{g}, h]} = e^{\beta[\delta_\emptyset, \bar{g}, h]} \sum_{C \in \mathcal{C}(Z_{k+1})} \prod_{C_i \in C} \beta[\delta_{C_i}, \bar{g}, h]$$

Using the derivative form (2.44), we can finally write:

$$v_{\Xi_{k+1|k+1}}(x_0 | Z_{1:k+1}) = \frac{\left[ \frac{\delta}{\delta x_0} \left( \frac{\delta}{\delta Z_{k+1}} e^{\beta[\delta_\emptyset, \bar{g}, h]} \right) \right]_{g^{1...S}=0, h=1}}{\left[ \frac{\delta}{\delta Z_{k+1}} e^{\beta[\delta_\emptyset, \bar{g}, h]} \right]_{g^{1...S}=0, h=1}} K_{\mathcal{X}}^{-1}$$
$$= \frac{\left[ \frac{\delta}{\delta x_0} \left( e^{\beta[\delta_\emptyset, \bar{g}, h]} \sum_{C \in \mathcal{C}(Z_{k+1})} \prod_{C_i \in C} \beta[\delta_{C_i}, \bar{g}, h] \right) \right]_{g^{1...S}=0, h=1}}{e^{\beta[\delta_\emptyset, 1]} \sum_{C \in \mathcal{C}(Z_{k+1})} \prod_{C_i \in C} \beta[\delta_{C_i}, 1]} K_{\mathcal{X}}^{-1}$$

Recall from proposition 2.5 that a cross-term vanishes when derivated in two points from the target space. Thus:

$$
\begin{aligned}
&v_{\Xi_{k+1|k+1}}(x_0|Z_{1:k+1}) \\
&= \frac{\left[ e^{\beta[\delta_\emptyset,\bar{g},h]} \beta[\delta_\emptyset,\bar{g},\delta_{x_0}] \sum_{C\in\mathcal{C}(Z_{k+1})} \prod_{C_i\in C} \beta[\delta_{C_i},\bar{g},h] \right]_{g^{1\dots S}=0,h=1}}{e^{\beta[\delta_\emptyset,1]} \sum_{C\in\mathcal{C}(Z_{k+1})} \prod_{C_i\in C} \beta[\delta_{C_i},1]} K_{\mathcal{X}}^{-1} \\
&\quad + \frac{\left[ e^{\beta[\delta_\emptyset,\bar{g},h]} \sum_{C\in\mathcal{C}(Z_{k+1})} \sum_{C_i\in C} \left( \beta[\delta_{C_i},\bar{g},\delta_{x_0}] \prod_{C_j\neq C_j} \beta[\delta_{C_j},\bar{g},h] \right) \right]_{g^{1\dots S}=0,h=1}}{e^{\beta[\delta_\emptyset,1]} \sum_{C\in\mathcal{C}(Z_{k+1})} \prod_{C_i\in C} \beta[\delta_{C_i},1]} K_{\mathcal{X}}^{-1} \\
&= \beta[\delta_\emptyset,\delta_{x_0}] K_{\mathcal{X}}^{-1} + \frac{\sum_{C\in\mathcal{C}(Z_{k+1})} \sum_{C_i\in C} \left( \beta[\delta_{C_i},\delta_{x_0}] \prod_{C_j\neq C_j} \beta[\delta_{C_j},1] \right)}{\sum_{C\in\mathcal{C}(Z_{k+1})} \prod_{C_i\in C} \beta[\delta_{C_i},1]} K_{\mathcal{X}}^{-1}
\end{aligned}
$$

□

## Proposition 2.6

*Proof.* Let $x_0 \in \mathcal{X}$ be any target state, $\{z^j\}_{j=1}^S$, $z^j \in \mathcal{Z}^j$ be any family of measurements, $J \subseteq [1\ S]$. Let $(S_{k+1}(p))_{p=1}^{P_{k+1}}$, $(T_{k+1}(p))_{p=0}^{P_{k+1}}$ be the joint partitioning at time $k+1$. Then, using equation (2.38) gives:

$$
\begin{aligned}
\beta[\delta_\emptyset,\bar{g},\delta_{x_0}] &= \prod_{j=1}^{S} (1 - p_{k+1}^{d,j}(x_0) + p_{k+1}^{d,j}(x_0) f_{k+1}^{o,j}[g^j|x_0]) v_{\Xi_{k+1|k}}(x_0|Z_{1:k}) K_{\mathcal{X}} \\
&= \prod_{q=1}^{P_{k+1}} \left( \prod_{j\in S_{k+1}(q)} (1 - p_{k+1}^{d,j}(x_0) + p_{k+1}^{d,j}(x_0) f_{k+1}^{o,j}[g^j|x_0]) \right) v_{\Xi_{k+1|k}}(x_0|Z_{1:k}) K_{\mathcal{X}}
\end{aligned}
$$

If $\exists p \in [1\ P], x_0 \in T_{k+1}(p)$ then:

$$\begin{aligned}
&\beta[\delta_\emptyset, \bar{g}, \delta_{x_0}] \\
&= \left( \prod_{j \in S_{k+1}(p)} (1 - p_{k+1}^{d,j}(x_0) + p_{k+1}^{d,j}(x_0) f_{k+1}^{o,j}[g^j|x_0]) \right) \\
&\quad \times \prod_{q \neq p} \left( \prod_{j \in S_{k+1}(q)} (1 - \underbrace{p_{k+1}^{d,j}(x_0)}_{=0} + \underbrace{p_{k+1}^{d,j}(x_0)}_{=0} f_{k+1}^{o,j}[g^j|x_0]) \right) v_{\Xi_{k+1|k}}(x_0|Z_{1:k}) K_{\mathcal{X}} \\
&= \prod_{j \in S_{k+1}(p)} (1 - p_{k+1}^{d,j}(x_0) + p_{k+1}^{d,j}(x_0) f_{k+1}^{o,j}[g^j|x_0]) v_{\Xi_{k+1|k}}(x_0|Z_{1:k}) K_{\mathcal{X}} \\
&= \beta_p[\delta_\emptyset, \bar{g}, \delta_{x_0}]
\end{aligned}$$

Otherwise, $x_0 \in T_{k+1}(0)$ and therefore:

$$\begin{aligned}
&\beta[\delta_\emptyset, \bar{g}, \delta_{x_0}] \\
&= \prod_{q=1}^{P_{k+1}} \left( \prod_{j \in S_{k+1}(q)} (1 - \underbrace{p_{k+1}^{d,j}(x_0)}_{=0} + \underbrace{p_{k+1}^{d,j}(x_0)}_{=0} f_{k+1}^{o,j}[g^j|x_0]) \right) v_{\Xi_{k+1|k}}(x_0|Z_{1:k}) K_{\mathcal{X}} \\
&= v_{\Xi_{k+1|k}}(x_0|Z_{1:k}) K_{\mathcal{X}}
\end{aligned}$$

Likewise, using equation (2.39) gives:

$$\begin{aligned}
&\beta[\delta_{\{z^j, j \in J\}}, \bar{g}, h] \\
&= \begin{cases}
\lambda_{k+1}^{c,j_0} c_{k+1}^{j_0}(z^{j_0}) K_{\mathcal{Z}^{j_0}} + v_{\Xi_{k+1|k}}[h p_{k+1}^{d,j_0} L_{k+1}^{z^{j_0},j_0} K_{\mathcal{Z}^{j_0}} \prod_{j \neq j_0} (1 - p_{k+1}^{d,j} + p_{k+1}^{d,j} f_{k+1}^{o,j}[g^j|.])] & (J = \{j_0\}) \\
v_{\Xi_{k+1|k}}[h \prod_{j \in J} (p_{k+1}^{d,j} L_{k+1}^{z^j,j} K_{\mathcal{Z}^j}) \prod_{j \notin J} (1 - p_{k+1}^{d,j} + p_{k+1}^{d,j} f_{k+1}^{o,j}[g^j|.])] & (|J| \geqslant 2)
\end{cases}
\end{aligned}$$

$$= \begin{cases} \lambda_{k+1}^{c,j_0} c_{k+1}^{j_0}(z^{j_0}) K_{\mathcal{Z}^{j_0}} + \left( \sum_{q=0}^{P_{k+1}} \int_{T_{k+1}(q)} h(x) p_{k+1}^{d,j_0}(x) L_{k+1}^{z^{j_0},j_0}(x) K_{\mathcal{Z}^{j_0}} \right. \\ \qquad \left. \times \prod_{j \neq j_0} (1 - p_{k+1}^{d,j}(x) + p_{k+1}^{d,j}(x) f_{k+1}^{o,j}[g^j|x])] \right) v_{\Xi_{k+1|k}}(x|Z_{1:k}) dx \\ \qquad\qquad (J = \{j_0\}) \\ \left( \sum_{q=0}^{P_{k+1}} \int_{T_{k+1}(q)} h(x) \prod_{j \in J} (p_{k+1}^{d,j}(x) L_{k+1}^{z^j,j}(x) K_{\mathcal{Z}^j}) \right. \\ \qquad \left. \times \prod_{j \notin J} (1 - p_{k+1}^{d,j}(x) + p_{k+1}^{d,j}(x) f_{k+1}^{o,j}[g^j|x]) \right) v_{\Xi_{k+1|k}}(x|Z_{1:k}) dx \\ \qquad\qquad (|J| \geqslant 2) \end{cases}$$

If $\exists p \in [1\ P_{k+1}], J \subseteq S_{k+1}(p)$ then:

$$
\begin{aligned}
&\beta[\delta_{\{z^j, j\in J\}}, \bar{g}, h] \\
&= \begin{cases}
\lambda_{k+1}^{c,j_0} c_{k+1}^{j_0}(z^{j_0}) K_{\mathcal{Z}^{j_0}} \\
\quad + \left( \int_{T_{k+1}(p)} h(x) p_{k+1}^{d,j_0}(x) L_{k+1}^{z^{j_0},j_0}(x) K_{\mathcal{Z}^{j_0}} \prod_{\substack{j\in S_{k+1}(p)\\ j\neq j_0}} (1 - p_{k+1}^{d,j}(x) + p_{k+1}^{d,j}(x) f_{k+1}^{o,j}[g^j|x]) \right. \\
\qquad \left. \times \prod_{j\notin S_{k+1}(p)} (1 - \underbrace{p_{k+1}^{d,j}(x)}_{=0} + \underbrace{p_{k+1}^{d,j}(x)}_{=0} f_{k+1}^{o,j}[g^j|x]) \right) v_{\Xi_{k+1|k}}(x|Z_{1:k}) dx \\
\quad + \left( \sum_{q\neq p} \int_{T_{k+1}(q)} h(x) \underbrace{p_{k+1}^{d,j_0}(x)}_{=0} L_{k+1}^{z^{j_0},j_0}(x) K_{\mathcal{Z}^{j_0}} \right. \\
\qquad \left. \times \prod_{j\neq j_0} (1 - p_{k+1}^{d,j}(x) + p_{k+1}^{d,j}(x) f_{k+1}^{o,j}[g^j|x]) \right) v_{\Xi_{k+1|k}}(x|Z_{1:k}) dx \\
\qquad\qquad (J = \{j_0\}) \\
\left( \int_{T_{k+1}(p)} h(x) \prod_{j\in J} (p_{k+1}^{d,j}(x) L_{k+1}^{z^j,j}(x) K_{\mathcal{Z}^j}) \prod_{\substack{j\in S_{k+1}(p)\\ j\notin J}} (1 - p_{k+1}^{d,j}(x) + p_{k+1}^{d,j}(x) f_{k+1}^{o,j}[g^j|x]) \right. \\
\qquad \left. \times \prod_{j\notin S_{k+1}(p)} (1 - \underbrace{p_{k+1}^{d,j}(x)}_{=0} + \underbrace{p_{k+1}^{d,j}(x)}_{=0} f_{k+1}^{o,j}[g^j|x]) \right) v_{\Xi_{k+1|k}}(x|Z_{1:k}) dx \\
\quad + \left( \sum_{q\neq p} \int_{T_{k+1}(q)} h(x) \prod_{j\in J} (\underbrace{p_{k+1}^{d,j}(x)}_{=0} L_{k+1}^{z^j,j}(x) K_{\mathcal{Z}^j}) \right. \\
\qquad \left. \times \prod_{j\notin J} (1 - p_{k+1}^{d,j}(x) + p_{k+1}^{d,j}(x) f_{k+1}^{o,j}[g^j|x]) \right) v_{\Xi_{k+1|k}}(x|Z_{1:k}) dx \\
\qquad\qquad (|J| \geqslant 2)
\end{cases}
\end{aligned}
$$

$$
= \begin{cases}
\lambda_{k+1}^{c,j_0} c_{k+1}^{j_0}(z^{j_0}) K_{\mathcal{Z}^{j_0}} + \left( \displaystyle\int_{T_{k+1}(p)} h(x) p_{k+1}^{d,j_0}(x) L_{k+1}^{z^{j_0},j_0}(x) K_{\mathcal{Z}^{j_0}} \right. \\
\qquad \left. \times \displaystyle\prod_{\substack{j \in S_{k+1}(p) \\ j \neq j_0}} (1 - p_{k+1}^{d,j}(x) + p_{k+1}^{d,j}(x) f_{k+1}^{o,j}[g^j|x]) \right) v_{\Xi_{k+1|k}}(x|Z_{1:k}) dx \\
\qquad\qquad (J = \{j_0\}) \\
\left( \displaystyle\int_{T_{k+1}(p)} h(x) \prod_{j \in J} (p_{k+1}^{d,j}(x) L_{k+1}^{z^j,j}(x) K_{\mathcal{Z}^j}) \right. \\
\qquad \left. \times \displaystyle\prod_{\substack{j \in S_{k+1}(p) \\ j \notin J}} (1 - p_{k+1}^{d,j}(x) + p_{k+1}^{d,j}(x) f_{k+1}^{o,j}[g^j|x]) \right) v_{\Xi_{k+1|k}}(x|Z_{1:k}) dx \\
\qquad\qquad (|J| \geqslant 2)
\end{cases}
$$

$$
= \begin{cases}
\lambda_{k+1}^{c,j_0} c_{k+1}^{j_0}(z^{j_0}) K_{\mathcal{Z}^{j_0}} \\
\qquad + v_{\Xi_{k+1|k}}[h 1_{T_{k+1}(p)} p_{k+1}^{d,j_0} L_{k+1}^{z^{j_0},j_0} K_{\mathcal{Z}^{j_0}} \displaystyle\prod_{\substack{j \in S_{k+1}(p) \\ j \neq j_0}} (1 - p_{k+1}^{d,j} + p_{k+1}^{d,j} f_{k+1}^{o,j}[g^j|.])] \\
\qquad\qquad (J = \{j_0\}) \\
v_{\Xi_{k+1|k}}[h 1_{T_{k+1}(p)} \displaystyle\prod_{j \in J} (p_{k+1}^{d,j} L_{k+1}^{z^j,j} K_{\mathcal{Z}^j}) \prod_{\substack{j \in S_{k+1}(p) \\ j \notin J}} (1 - p_{k+1}^{d,j} + p_{k+1}^{d,j} f_{k+1}^{o,j}[g^j|.])] \\
\qquad\qquad (|J| \geqslant 2)
\end{cases}
$$

$$
= \beta_p[\delta_{\{z^j, j \in J\}}, \bar{g}, h]
$$

Otherwise $\exists p_1, p_2 \in [1\ P_{k+1}], p_1 \neq p_2, \exists j_1, j_2 \in J, j_1 \neq j_2, j_1 \in S_{k+1}(p_1), j_2 \in S_{k+1}(p_2)$ and therefore:

$$
\begin{aligned}
&\beta[\delta_{\{z^j, j\in J\}}, \bar{g}, h] \\
&= \Bigg( \int_{T_{k+1}(p_1)} h(x) \underbrace{p_{k+1}^{d,j_2}(x)}_{=0} L_{k+1}^{z^{j_2},j_2}(x) \prod_{\substack{j\in J \\ j\neq j_2}} \Big( p_{k+1}^{d,j}(x) L_{k+1}^{z^j,j}(x) \Big) \\
&\qquad \prod_{j\notin J} \Big( 1 - p_{k+1}^{d,j}(x) + p_{k+1}^{d,j}(x) f_{k+1}^{o,j}[g^j|x] \Big) \Bigg) v_{\Xi_{k+1|k}}(x|Z_{1:k}) dx \\
&\quad + \Bigg( \int_{\substack{T_{k+1}(q) \\ q\neq p_1}} h(x) \underbrace{p_{k+1}^{d,j_1}(x)}_{=0} L_{k+1}^{z^{j_1},j_1}(x) \prod_{\substack{j\in J \\ j\neq j_1}} \Big( p_{k+1}^{d,j}(x) L_{k+1}^{z^j,j}(x) \Big) \\
&\qquad \prod_{j\notin J} \Big( 1 - p_{k+1}^{d,j}(x) + p_{k+1}^{d,j}(x) f_{k+1}^{o,j}[g^j|x] \Big) \Bigg) v_{\Xi_{k+1|k}}(x|Z_{1:k}) dx \\
&= 0
\end{aligned}
$$

Likewise, using equation (2.40) gives:

$$
\begin{aligned}
&\beta[\delta_{\{z^j, j\in J\}}, \bar{g}, \delta_{x_0}] \\
&= \Bigg( \prod_{j\in J} \Big( p_{k+1}^{d,j}(x_0) L_{k+1}^{z^j,j}(x_0) \Big) \prod_{j\notin J} \Big( 1 - p_{k+1}^{d,j}(x_0) + p_{k+1}^{d,j}(x_0) f_{k+1}^{o,j}[g^j|x_0] \Big) \Bigg) \\
&\qquad\qquad v_{\Xi_{k+1|k}}(x_0|Z_{1:k}) K_{\mathcal{X}}
\end{aligned}
$$

If $\exists p \in [1\ P_{k+1}], J \subseteq S_{k+1}(p), x_0 \in T_{k+1}(p)$ then:

$$
\begin{aligned}
&\beta[\delta_{\{z^j, j\in J\}}, \bar{g}, \delta_{x_0}] \\
&= \Bigg( \prod_{j\in J} \Big( p_{k+1}^{d,j}(x_0) L_{k+1}^{z^j,j}(x_0) \Big) \prod_{\substack{j\in S_{k+1}(p) \\ j\notin J}} \Big( 1 - p_{k+1}^{d,j}(x_0) + p_{k+1}^{d,j}(x_0) f_{k+1}^{o,j}[g^j|x_0] \Big) \\
&\qquad \prod_{\substack{j\in S_{k+1}(p) \\ j\notin J}} \Big( 1 - \underbrace{p_{k+1}^{d,j}(x_0)}_{=0} + \underbrace{p_{k+1}^{d,j}(x_0)}_{=0} f_{k+1}^{o,j}[g^j|x_0] \Big) \Bigg) v_{\Xi_{k+1|k}}(x_0|Z_{1:k}) K_{\mathcal{X}} \\
&= \Bigg( \prod_{j\in J} \Big( p_{k+1}^{d,j}(x_0) L_{k+1}^{z^j,j}(x_0) \Big) \prod_{\substack{j\in S_{k+1}(p) \\ j\notin J}} \Big( 1 - p_{k+1}^{d,j}(x_0) + p_{k+1}^{d,j}(x_0) f_{k+1}^{o,j}[g^j|x_0] \Big) \Bigg) \\
&\qquad\qquad v_{\Xi_{k+1|k}}(x_0|Z_{1:k}) K_{\mathcal{X}} \\
&= \beta_p[\delta_{\{z^j, j\in J\}}, \bar{g}, \delta_{x_0}]
\end{aligned}
$$

Otherwise, either $x_0 \in T_{k+1}(0)$ or $\exists p_1, p_2 \in [1\ P_{k+1}], p_1 \neq p_2, \exists j_0 \in J, x \in T_{k+1}(p_1), j_0 \in S_{k+1}(p_2)$ and in both cases $p_{k+1}^{d,j_0}(x_0)$ so that:

$$\begin{aligned}
&\beta[\delta_{\{z^j, j\in J\}}, \bar{g}, \delta_{x_0}] \\
&= \Bigg( \underbrace{p_{k+1}^{d,j_0}(x_0)}_{=0} L_{k+1}^{z^{j_0},j_0}(x_0) \prod_{\substack{j\in J \\ j\neq j_0}} \left( p_{k+1}^{d,j}(x_0) L_{k+1}^{z^j,j}(x_0) \right) \\
&\qquad \prod_{j\notin J} \left( 1 - p_{k+1}^{d,j}(x_0) + p_{k+1}^{d,j}(x_0) f_{k+1}^{o,j}[g^j|x_0] \right) \Bigg) v_{\Xi_{k+1|k}}(x_0|Z_{1:k}) K_{\mathcal{X}} \\
&= 0
\end{aligned}$$

□

## Theorem 2.5

*Proof.* Let $(S_{k+1}(p))_{p=1}^{P_{k+1}}$, $(T_{k+1}(p))_{p=0}^{P_{k+1}}$ be the current joint partitioning given by definition 2.9, and $x \in \mathcal{X}$ any target state. The first case to cover is when $x \in T_{k+1}(0)$, i.e. $x$ is outside every FOV. Using theorem 2.4 and proposition 2.6 gives immediately:

$$\begin{aligned}
&v_{\Xi_{k+1|k+1}}(x|Z_{1:k+1}) \\
&= \underbrace{\beta[\delta_\emptyset, \delta_x]}_{=v_{\Xi_{k+1|k}}(x|Z_{1:k})K_{\mathcal{X}}} K_{\mathcal{X}}^{-1} + \frac{\displaystyle\sum_{C\in\mathcal{C}(Z_{k+1})} \sum_{C_i\in C} \left( \underbrace{\beta[\delta_{C_i}, \delta_x]}_{=0} \prod_{C_j\neq C_j} \beta[\delta_{C_j}, 1] \right)}{\displaystyle\sum_{C\in\mathcal{C}(Z_{k+1})} \prod_{C_i\in C} \beta[\delta_{C_i}, 1]} K_{\mathcal{X}}^{-1} \\
&= v_{\Xi_{k+1|k}}(x|Z_{1:k})
\end{aligned}$$

Now, assume that $x \in T_{k+1}(p), p \neq 0$. We must show that:

$$v_{\Xi_{k+1|k}}(x|Z_{1:k}) = \beta_p[\delta_\emptyset, \delta_x] K_{\mathcal{X}}^{-1} + \frac{\displaystyle\sum_{C\in\mathcal{C}(Z_{k+1}^{(p)})} \sum_{C_i\in C} \left( \beta_p[\delta_{C_i}, \delta_x] \prod_{C_j\neq C_j} \beta_p[\delta_{C_j}, 1] \right)}{\displaystyle\sum_{C\in\mathcal{C}(Z_{k+1}^{(p)})} \prod_{C_i\in C} \beta_p[\delta_{C_i}, 1]} K_{\mathcal{X}}^{-1} \tag{41}$$

Note that we need to prove equation (41) for $P_{k+1} = 1$ and $P_{k+1} = 2$ only. Indeed, results from proposition 2.6 clearly hold for any sensor partition coarser than $(S_{k+1}(p))_{p=1}^{P_{k+1}}$. Thus, if $P_{k+1} > 2$, applying result (41) to any coarser partition of

$(S_{k+1}(p))_{p=1}^{P_{k+1}}$ with two elements, namely $(S^c_{k+1}(1), S^c_{k+1}(2))$, and using (41) twice, once on the restriction of $\mathcal{X}$ to $T^c_{k+1}(1)$ with a two-element partition of $S^c_{k+1}(1)$ and once on the restriction of $\mathcal{X}$ to $T^c_{k+1}(2)$ with a two-element partition of $S^c_{k+1}(2)$, and proceeding with finer and finer partitions up to $(S_{k+1}(p))_{p=1}^{P_{k+1}}$ yields the result for any partition size $P_{k+1}$.

The case $P_{k+1} = 1$ is straightforward using theorem 2.4. Assume that $P_{k+1} = 2$, either $x \in T_{k+1}(1)$ or $x \in T_{k+1}(2)$. Without loss of generality, assume that $x \in T_{k+1}(1)$. For any $C \in \mathcal{C}(Z_{k+1})$ and any $C_i \in C$, according to proposition 2.6:

$$\beta[\delta_{C_i}, \bar{g}, h] = \begin{cases} \beta_1[\delta_{C_i}, \bar{g}, h] & (C_i \in \mathcal{M}(Z^{(1)}_{k+1})) \\ \beta_2[\delta_{C_i}, \bar{g}, h] & (C_i \in \mathcal{M}(Z^{(2)}_{k+1})) \\ 0 & (otherwise) \end{cases}$$

Therefore we can write:

$$\sum_{C \in \mathcal{C}(Z_{k+1})} \prod_{C_i \in C} \beta[\delta_{C_i}, 1] = \sum_{\substack{C \in \mathcal{C}(Z_{k+1}) \\ C = C^{(1)} \cup C^{(2)} \\ C^{(1)} \subseteq \mathcal{M}(Z^{(1)}_{k+1}) \\ C^{(2)} \subseteq \mathcal{M}(Z^{(2)}_{k+1})}} \left( \prod_{C_i \in C^{(1)}} \beta_1[\delta_{C_i}, 1] \right) \left( \prod_{C_i \in C^{(2)}} \beta_2[\delta_{C_i}, 1] \right)$$

Since $Z_{k+1} = Z^{(1)}_{k+1} \sqcup Z^{(2)}_{k+1}$, using equation (2.49) gives:

$$\begin{aligned} \sum_{C \in \mathcal{C}(Z_{k+1})} \prod_{C_i \in C} \beta[\delta_{C_i}, 1] &= \sum_{\substack{C = C^{(1)} \cup C^{(2)} \\ C^{(1)} \in \mathcal{C}(Z^{(1)}_{k+1}) \\ C^{(2)} \in \mathcal{C}(Z^{(2)}_{k+1})}} \left( \prod_{C_i \in C^{(1)}} \beta_1[\delta_{C_i}, 1] \right) \left( \prod_{C_i \in C^{(2)}} \beta_2[\delta_{C_i}, 1] \right) \\ &= \left( \sum_{C \in \mathcal{C}(Z^{(1)}_{k+1})} \prod_{C_i \in C} \beta_1[\delta_{C_i}, 1] \right) \left( \sum_{C \in \mathcal{C}(Z^{(2)}_{k+1})} \prod_{C_i \in C} \beta_2[\delta_{C_i}, 1] \right) \end{aligned} \quad (42)$$

Likewise, the numerator in data update equation (2.53) can be simplified:

$$
\begin{aligned}
&\sum_{C\in\mathcal{C}(Z_{k+1})}\sum_{C_i\in C}\left(\beta[\delta_{C_i},\delta_x]\prod_{C_j\neq C_j}\beta[\delta_{C_j},1]\right)\\
&=\sum_{\substack{C\in\mathcal{C}(Z_{k+1})\\ C=C^{(1)}\cup C^{(2)}\\ C^{(1)}\subseteq\mathcal{M}(Z_{k+1}^{(1)})\\ C^{(2)}\subseteq\mathcal{M}(Z_{k+1}^{(2)})}}\sum_{C_i\in C^{(1)}}\left(\beta[\delta_{C_i},\delta_x]\left(\prod_{\substack{C_j\in C^{(1)}\\ C_j\neq C_i}}\beta_1[\delta_{C_j},1]\right)\left(\prod_{\substack{C_k\in C^{(2)}\\ C_k\neq C_i}}\beta_2[\delta_{C_k},1]\right)\right)\\
&=\sum_{\substack{C=C^{(1)}\cup C^{(2)}\\ C^{(1)}\in\mathcal{C}(Z_{k+1}^{(1)})\\ C^{(2)}\in\mathcal{C}(Z_{k+1}^{(2)})}}\sum_{C_i\in C^{(1)}}\left(\beta_1[\delta_{C_i},1]\prod_{\substack{C_j\in C^{(1)}\\ C_j\neq C_i}}\beta_1[\delta_{C_j},1]\right)\left(\prod_{C_k\in C^{(2)}}\beta_2[\delta_{C_k},1]\right)\\
&\quad+\sum_{\substack{C=C^{(1)}\cup C^{(2)}\\ C^{(1)}\in\mathcal{C}(Z_{k+1}^{(1)})\\ C^{(2)}\in\mathcal{C}(Z_{k+1}^{(2)})}}\sum_{C_i\in C^{(2)}}\left(\prod_{C_k\in C^{(1)}}\beta_1[\delta_{C_k},1]\right)\left(\underbrace{\beta_2[\delta_{C_1},\delta_x]}_{=0}\prod_{\substack{C_j\in C^{(2)}\\ C_j\neq C_i}}\beta_2[\delta_{C_j},1]\right)\\
&=\left(\sum_{C\in\mathcal{C}(Z_{k+1}^{(1)})}\sum_{C_i\in C}\left(\beta_1[\delta_{C_i},\delta_x]\prod_{C_j\neq C_i}\beta_1[\delta_{C_j},1]\right)\right)\left(\sum_{C\in\mathcal{C}(Z_{k+1}^{(2)})}\prod_{C_i\in C}\beta_2[\delta_{C_i},1]\right)\qquad(43)
\end{aligned}
$$

Thus, substituting the simplified expressions of the numerator (43) and denominator (42) in equation (2.53) yields:

$$
\begin{aligned}
&v_{\Xi_{k+1|k+1}}(x|Z_{1:k+1})\\
&=\beta[\delta_\emptyset,\delta_x]K_{\mathcal{X}}^{-1}\\
&\quad+\frac{\left(\sum_{C\in\mathcal{C}(Z_{k+1}^{(1)})}\sum_{C_i\in C}\left(\beta_1[\delta_{C_i},\delta_x]\prod_{C_j\neq C_i}\beta_1[\delta_{C_j},1]\right)\right)\left(\sum_{C\in\mathcal{C}(Z_{k+1}^{(2)})}\prod_{C_i\in C}\beta_2[\delta_{C_i},1]\right)}{\left(\sum_{C\in\mathcal{C}(Z_{k+1}^{(1)})}\prod_{C_i\in C}\beta_1[\delta_{C_i},1]\right)\left(\sum_{C\in\mathcal{C}(Z_{k+1}^{(2)})}\prod_{C_i\in C}\beta_2[\delta_{C_i},1]\right)}K_{\mathcal{X}}^{-1}
\end{aligned}
$$

According to equation (2.60), since $x \in T_{k+1}(1)$, $\beta[\delta_\emptyset, \delta_x] = \beta_1[\delta_\emptyset, \delta_x]$ and therefore:

$$v_{\Xi_{k+1|k+1}}(x|Z_{1:k+1}) = \beta_1[\delta_\emptyset, \delta_x]K_{\mathcal{X}}^{-1} + \frac{\sum_{C \in \mathcal{C}(Z_{k+1}^{(1)})} \sum_{C_i \in C} \left( \beta_1[\delta_{C_i}, \delta_x] \prod_{C_j \neq C_i} \beta_1[\delta_{C_j}, 1] \right)}{\sum_{C \in \mathcal{C}(Z_{k+1}^{(1)})} \prod_{C_i \in C} \beta_1[\delta_{C_i}, 1]} K_{\mathcal{X}}^{-1}$$

□

# Chapter 3: Multi-sensor management within PHD framework

## Proposition 3.2

This proof is an extension of Mahler's in the single-sensor case (see [Mahl 04]).

*Proof.* Let $u \in U_{k+1}$ be any available control and $x \in \mathcal{X}$ any target state. Using the definition of the predictive PHD (definition 3.3) with the PIMS as the predictive observation RFS gives:

$$v_{\Xi^u_{k+1|k+1}}(x|Z_{1:k}) = \mathbb{E}_\omega[v_{\Xi_{k+1|k+1}}(.|Z_{1:k} \cup \Sigma^{WE}_u(\omega))]$$

Which simplifies, by construction of the PIMS (proposition 3.1 and definition 3.5):

$$v_{\Xi^u_{k+1|k+1}}(x|Z_{1:k}) = \sum_{Z \subseteq Z^{WE}_{k+1}} p_{\Sigma^{WE}_u}(Z) v_{\Xi_{k+1|k+1}}(.|Z_{1:k} \cup Z)$$

Which gives, using the data update equation (2.53):

$$\begin{aligned}
&v_{\Xi^u_{k+1|k+1}}(x|Z_{1:k}) \\
&= \sum_{Z \subseteq Z^{WE}_{k+1}} p_{\Sigma^{WE}_u}(Z) \left( \beta[\delta_\emptyset, \delta_x] K_{\mathcal{X}}^{-1} + \frac{\sum_{C \in \mathcal{C}(Z)} \sum_{C_i \in C} \left( \beta[\delta_{C_i}, \delta_x] \prod_{C_j \neq C_j} \beta[\delta_{C_j}, 1] \right)}{\sum_{C \in \mathcal{C}(Z)} \prod_{C_i \in C} \beta[\delta_{C_i}, 1]} K_{\mathcal{X}}^{-1} \right) \\
&= \underbrace{\left( \sum_{Z \subseteq Z^{WE}_{k+1}} p_{\Sigma^{WE}_u}(Z) \right)}_{=1} \beta[\delta_\emptyset, \delta_x] K_{\mathcal{X}}^{-1} \\
&\quad + \sum_{Z \subseteq Z^{WE}_{k+1}} p_{\Sigma^{WE}_u}(Z) \frac{\sum_{C \in \mathcal{C}(Z)} \sum_{C_i \in C} \left( \beta[\delta_{C_i}, \delta_x] \prod_{C_j \neq C_j} \beta[\delta_{C_j}, 1] \right)}{\sum_{C \in \mathcal{C}(Z)} \prod_{C_i \in C} \beta[\delta_{C_i}, 1]} K_{\mathcal{X}}^{-1} \\
&= \beta[\delta_\emptyset, \delta_x] K_{\mathcal{X}}^{-1} + \sum_{Z \subseteq Z^{WE}_{k+1}} p_{\Sigma^{WE}_u}(Z) \frac{\sum_{C \in \mathcal{C}(Z)} \sum_{C_i \in C} \left( \beta[\delta_{C_i}, \delta_x] \prod_{C_j \neq C_j} \beta[\delta_{C_j}, 1] \right)}{\sum_{C \in \mathcal{C}(Z)} \prod_{C_i \in C} \beta[\delta_{C_i}, 1]} K_{\mathcal{X}}^{-1}
\end{aligned}$$

□

## Corollary 3.1

*Proof.* Let $u \in U_{K+1}$ be any available control and $x \in \mathcal{X}$ any target state. In the same way as in the proof of theorem 2.2, the data update equation (3.17) is simplified in the single-sensor as follows:

$$v_{\Xi^u_{k+1|k+1}}(x|Z_{1:k}) = \beta[\delta_\emptyset, \delta_x]K_{\mathcal{X}}^{-1} + \sum_{Z \subseteq Z_{k+1}^{WE}} p_{\Sigma_u^{WE}}(Z) \sum_{z \in Z} \frac{\beta[\delta_{\{z\}}, \delta_x]}{\beta[\delta_{\{z\}}, 1]} K_{\mathcal{X}}^{-1} \tag{44}$$

Let $\{z_1, ..., z_M\} = Z_{k+1}^{WE}$ be the ideal measurements. For any $1 \leqslant m \leqslant M$, let $p^m$ be the restriction of $p_{\Sigma_u^{WE}}$ to the first $m$ measurements in $Z_{k+1}^{WE}$. That is, $p^m$ is the function defined on any subset $Z \subseteq \{z_1, ..., z_m\}$ by:

$$p^m(Z) = \prod_{z \in Z} p_u^d((\rho_{k+1})^{-1}(z)) \prod_{z \in \{z_1,...,z_m\} \setminus Z} \left(1 - p_u^d((\rho_{k+1})^{-1}(z))\right)$$

Then, let us prove by induction on $m$ that:

$$\sum_{Z \subseteq \{z_1,...,z_m\}} p^m(Z) \sum_{z \in Z} \frac{\beta[\delta_{\{z\}}, \delta_x]}{\beta[\delta_{\{z\}}, 1]} = \sum_{z \in \{z_1,...,z_M\}} p_u^d((\rho_{k+1})^{-1}(z)) \frac{\beta[\delta_{\{z\}}, \delta_x]}{\beta[\delta_{\{z\}}, 1]} \tag{45}$$

Let us consider the base case $m = 1$:

$$\sum_{Z \subseteq \{z_1\}} p^1(Z) \sum_{z \in Z} \frac{\beta[\delta_{\{z\}}, \delta_x]}{\beta[\delta_{\{z\}}, 1]} = p^1(\emptyset) \times 0 + p^1(\{z_1\}) \frac{\beta[\delta_{\{z\}}, \delta_x]}{\beta[(\delta_{\{z\}}, 1]} = p_u^d((\rho_{k+1})^{-1}(z_1)) \frac{\beta[\delta_{\{z\}}, \delta_x]}{\beta[\delta_{\{z\}}, 1]}$$

Assuming that case $m < M$ is true, let us prove that case $m + 1$ is true:

$$\begin{aligned}
&\sum_{Z \subseteq \{z_1,...,z_{m+1}\}} p^{m+1}(Z) \sum_{z \in Z} \frac{\beta[\delta_{\{z\}}), \delta_x]}{\beta[\delta_{\{z\}}, 1]} \\
&= \sum_{Z \subseteq \{z_1,...,z_m\}} p^{m+1}(Z) \sum_{z \in Z} \frac{\beta[\delta_{\{z\}}, \delta_x]}{\beta[\delta_{\{z\}}, 1]} + \sum_{\substack{Z' = Z \cup \{z_{m+1}\} \\ Z \subseteq \{z_1,...,z_m\}}} p^{m+1}(Z') \sum_{z \in Z'} \frac{\beta[\delta_{\{z\}}, \delta_x]}{\beta[\delta_{\{z\}}, 1]} \\
&= \sum_{Z \subseteq \{z_1,...,z_m\}} p^m(Z) \left(1 - p_u^d((\rho_{k+1})^{-1}(z_{m+1}))\right) \sum_{z \in Z} \frac{\beta[\delta_{\{z\}}, \delta_x]}{\beta[\delta_{\{z\}}, 1]} \\
&\quad + \sum_{Z \subseteq \{z_1,...,z_m\}} p^m(Z) p_u^d((\rho_{k+1})^{-1}(z_{m+1})) \left( \sum_{z \in Z} \frac{\beta[\delta_{\{z\}}, \delta_x]}{\beta[\delta_{\{z\}}, 1]} + \frac{\beta[\delta_{\{z_{m+1}\}}, \delta_x]}{\beta[\delta_{\{z_{m+1}\}}, 1]} \right) \\
&= \sum_{Z \subseteq \{z_1,...,z_m\}} p^m(Z) \underbrace{\left(1 - p_u^d((\rho_{k+1})^{-1}(z_{m+1})) + p_u^d((\rho_{k+1})^{-1}(z_{m+1}))\right)}_{=1} \sum_{z \in Z} \frac{\beta[\delta_{\{z\}}, \delta_x]}{\beta[\delta_{\{z\}}, 1]} \\
&\quad + \underbrace{\left( \sum_{Z \subseteq \{z_1,...,z_m\}} p^m(Z) \right)}_{=1} p_u^d((\rho_{k+1})^{-1}(z_{m+1})) \frac{\beta[\delta_{\{z_{m+1}\}}, \delta_x]}{\beta[\delta_{\{z_{m+1}\}}, 1]}
\end{aligned}$$

Which gives, using (45) at step $m$:

$$\begin{aligned}
&\sum_{Z\subseteq\{z_1,\ldots,z_{m+1}\}} p^{m+1}(Z)\sum_{z\in Z}\frac{\beta[\delta_{\{z\}},\delta_x]}{\beta[\delta_{\{z\}},1]}\\
&=\sum_{z\in\{z_1,\ldots,z_m\}} p_u^d((\rho_{k+1})^{-1}(z))\frac{\beta[\delta_{\{z\}},\delta_x]}{\beta[\delta_{\{z\}},1]}+p_u^d((\rho_{k+1})^{-1}(z_{m+1}))\frac{\beta[\delta_{\{z_{m+1}\}},\delta_x]}{\beta[\delta_{\{z_{m+1}\}},1]}\\
&=\sum_{z\in\{z_1,\ldots,z_{m+1}\}} p_u^d((\rho_{k+1})^{-1}(z))\frac{\beta[\delta_{\{z\}},\delta_x]}{\beta[\delta_{\{z\}},1]}
\end{aligned}$$

Therefore, the case holds at step $m+1$ and the induction is true for any $m\leqslant M$. Combining the case at final step $M$ and expression (44) above yields:

$$v_{\Xi^u_{k+1|k+1}}(x|Z_{1:k})=\beta[\delta_\emptyset,\delta_x]K_{\mathcal{X}}^{-1}+\sum_{z\in Z^{WE}_{k+1}} p_u^d((\rho_{k+1})^{-1}(z))\frac{\beta[\delta_{\{z\}},\delta_x]}{\beta[\delta_{\{z\}},1]}K_{\mathcal{X}}^{-1}$$

That is, using the expressions of the derivated cross-terms given in proposition 2.3:

$$\begin{aligned}
&v_{\Xi^u_{k+1|k+1}}(x|Z_{1:k})\\
&=\left(1-p_u^d(x)+\sum_{z\in Z^{WE}_{k+1}} p_u^d((\rho_{k+1})^{-1}(z))\frac{p_u^d(x)L^z_{k+1}(x)}{\lambda^c_{k+1}c_{k+1}(z)+v_{\Xi_{k+1|k}}[p_u^dL^z_{k+1}]}\right)v_{\Xi_{k+1|k}}(x|Z_{1:k})
\end{aligned}$$

□

## Proposition 3.3

*Proof.* Let $u\in U_{K+1}$ be any available control and $x\in\mathcal{X}$ any target state. Using the definition of the predictive PHD (definition 3.3) with the PIMS as the predictive observation RFS gives:

$$v_{\Xi^u_{k+1|k+1}}(x|Z_{1:k})=\mathbb{E}_\omega[v_{\Xi_{k+1|k+1}}(.|Z_{1:k}\cup\Sigma^{WE}_u(\omega))]$$

Which simplifies, by construction of the PIMS (proposition 3.1 and definition 3.5):

$$v_{\Xi^u_{k+1|k+1}}(x|Z_{1:k})=\sum_{Z\subseteq Z^{WE}_{k+1}} p_{\Sigma^{WE}_u}(Z)v_{\Xi_{k+1|k+1}}(.|Z_{1:k}\cup Z)$$

Which gives, using the data update equation (2.53):

$$
\begin{aligned}
&v_{\Xi^u_{k+1|k+1}}(x|Z_{1:k}) \\
&= \sum_{Z\subseteq Z^{WE}_{k+1}} p_{\Sigma^{WE}_u}(Z) \begin{cases} v_{\Xi_{k+1|k}}(x|Z_{1:k}) & (x \in T_u(0)) \\ \beta_p[\delta_\emptyset, \delta_x]K_{\mathcal{X}}^{-1} + \dfrac{\sum\limits_{C\in\mathcal{C}(Z^{(p)})}\sum\limits_{C_i\in C}\left(\beta_p[\delta_{C_i},\delta_x]\prod\limits_{C_j\neq C_j}\beta_p[\delta_{C_j},1]\right)}{\sum\limits_{C\in\mathcal{C}(Z^{(p)}_{k+1})}\prod\limits_{C_i\in C}\beta_p[\delta_{C_i},1]}K_{\mathcal{X}}^{-1} & (x \in T_u(p), p\neq 0) \end{cases} \\
&= \begin{cases} \underbrace{\left(\sum\limits_{Z\subseteq Z^{WE}_{k+1}} p_{\Sigma^{WE}_u}(Z)\right)}_{=1} v_{\Xi_{k+1|k}}(x|Z_{1:k}) & (x \in T_u(0)) \\ \underbrace{\left(\sum\limits_{Z\subseteq Z^{WE}_{k+1}} p_{\Sigma^{WE}_u}(Z)\right)}_{=1} \beta_p[\delta_\emptyset, \delta_x]K_{\mathcal{X}}^{-1} \\ \quad + \underbrace{\sum\limits_{Z\subseteq Z^{WE}_{k+1}} p_{\Sigma^{WE}_u}(Z) \underbrace{\dfrac{\sum\limits_{C\in\mathcal{C}(Z^{(p)})}\sum\limits_{C_i\in C}\left(\beta_p[\delta_{C_i},\delta_x]\prod\limits_{C_j\neq C_j}\beta_p[\delta_{C_j},1]\right)}{\sum\limits_{C\in\mathcal{C}(Z^{(p)}_{k+1})}\prod\limits_{C_i\in C}\beta_p[\delta_{C_i},1]}}_{=f(Z,p)}}_{=T} K_{\mathcal{X}}^{-1} & (x \in T_u(p), p\neq 0) \end{cases}
\end{aligned}
$$

Let $Z$ be any subset of the ideal measurement set $Z^{WE}_{k+1}$. Then, according to definition 3.6:

$$
\begin{aligned}
&Z \cap \left(Z^{WE}_{k+1} \setminus Z^{WE}_u\right) \neq \emptyset \\
&\Rightarrow Z \cap \left(Z^{WE}_{k+1} \setminus \bigcup_{q=1}^{P_u} Z^{WE}_{u,q}\right) \neq \emptyset \\
&\Rightarrow \exists z \in Z \mid \exists j \in [1\ S], j \in S_u(q), z \in \mathcal{Z}^j, (\rho^j_{k+1})^{-1}(z) \in T_u(r), q \neq r \\
&\Rightarrow \exists z \in Z \mid \exists j \in [1\ S], p^{d,j}_u((\rho^j_{k+1})^{-1}(z)) = 0
\end{aligned}
$$

Which implies, according to the construction of the PIMS (proposition 3.1):

$$\Rightarrow p_{\Sigma_u^{WE}}(Z) = 0$$

Therefore, we can write:

$$\begin{aligned} T &= \sum_{Z \subseteq \bigsqcup_{q=1}^{P_u} Z_{u,q}^{WE}} p_{\Sigma_u^{WE}}(Z) f(Z,p) \\ &= \sum_{\substack{Z = A \sqcup B \\ A \subseteq Z_{u,p}^{WE} \\ B \subseteq \bigsqcup_{q \neq p} Z_{u,q}^{WE}}} p_{\Sigma_u^{WE}}(Z) f(Z,p) \end{aligned}$$

Since $\mathcal{C}(Z^{(p)})$ is the set of combinational terms based on measurements produced by sensors from partition element $S_u(p)$ only, $f(Z,p)$ equals $f(A,p)$ and thus:

$$T = \sum_{\substack{Z = A \sqcup B \\ A \subseteq Z_{u,p}^{WE} \\ B \subseteq \bigsqcup_{q \neq p} Z_{u,q}^{WE}}} p_{\Sigma_u^{WE}}(Z) f(A,p)$$

That is, according to the construction of the PIMS (proposition 3.1) with $A = \bigsqcup_{j \in S_u(p)} A_p^j$ and $B = \bigsqcup_{q \neq p} \bigsqcup_{j \in S_u(q)} B_q^j$:

$$\begin{aligned} T = & \sum_{\substack{Z = A \sqcup B \\ A \subseteq Z_{u,p}^{WE} \\ B \subseteq \sqcup_{q \neq p} Z_{u,q}^{WE}}} f(A,p) \\ & \times \underbrace{\prod_{j \in S_u(p)} \left( \prod_{z \in A^j} \left( p_u^{d,j}((\rho_{k+1}^j)^{-1}(z)) \right) \prod_{z \in Z_{u,p}^{WE,j} \setminus A^j} \left( 1 - p_u^{d,j}((\rho_{k+1}^j)^{-1}(z)) \right) \right)}_{= p_{\Sigma_{u,p}^{WE}}(A)} \\ & \times \prod_{q \neq p} \underbrace{\left( \prod_{j \in S_u(q)} \left( \prod_{z \in B^j} \left( p_u^{d,j}((\rho_{k+1}^j)^{-1}(z)) \right) \prod_{z \in Z_{u,q}^{WE,j} \setminus B^j} \left( 1 - p_u^{d,j}((\rho_{k+1}^j)^{-1}(z)) \right) \right) \right)}_{= p_{\Sigma_{u,q}^{WE}}(B_q)} \\ = & \left( \sum_{A \subseteq Z_{u,p}^{WE}} p_{\Sigma_{u,p}^{WE}}(A) f(A,p) \right) \left( \prod_{q \neq p} \underbrace{\left( \sum_{B_q \subseteq Z_{u,q}^{WE}} p_{\Sigma_{u,q}^{WE}}(B_q) \right)}_{=1} \right) \\ = & \sum_{A \subseteq Z_{u,p}^{WE}} p_{\Sigma_{u,p}^{WE}}(A) f(A,p) \end{aligned}$$

Substituting the new expression of $T$ in $v_{\Xi^u_{k+1|k+1}}(x)$ gives:

$$v_{\Xi^u_{k+1|k+1}}(x|Z_{1:k})$$
$$= \begin{cases} v_{\Xi_{k+1|k}}(x|Z_{1:k}) & (x \in T_u(0)) \\ \beta_p[\delta_\emptyset, \delta_x]K_{\mathcal{X}}^{-1} + \sum\limits_{Z \subseteq Z^{WE}_{u,p}} p_{\Sigma^{WE}_{u,p}}(Z) \dfrac{\sum\limits_{C \in \mathcal{C}(Z)} \sum\limits_{C_i \in C} \left( \beta_p[\delta_{C_i}, \delta_x] \prod\limits_{C_j \neq C_j} \beta_p[\delta_{C_j}, 1] \right)}{\sum\limits_{C \in \mathcal{C}(Z)} \prod\limits_{C_i \in C} \beta_p[\delta_{C_i}, 1]} K_{\mathcal{X}}^{-1} & \\ & (x \in T_u(p), p \neq 0) \end{cases}$$

□

# Appendix B: Importance sampling

This appendix focuses on the SMC implementation of a single-target tracking problem in a Bayesian framework. It is mainly based on two papers [Gewe 89, Douc 00] on sequential Monte Carlo sampling techniques.

## General description of the tracking problem

Assume that there is a single target evolving through time in the state space $\mathcal{X}$, whose state (position, velocity, etc.) is of interest. The sequence of target states $\{x_k, k \in \mathbb{N}\}$, $x_k \in \mathcal{X}$, is assumed to be an hidden Markov process with an initial distribution $t(x_0)$ (or $t(x_0|x_{-1})$ for notational convenience). The sequential observation of the target produces a sequence of measurements $\{z_k, k \in \mathbb{N}\}$, $z_k \in \mathcal{Z}$, assumed to be conditionally independent given the sequence of target states $\{x_k, k \in \mathbb{N}\}$. The target and measurement processes are completely described by the sequences of probability densities:

$$\{t(x_k|x_{k-1}), k \in \mathbb{N}\} \tag{46}$$
$$\{g(z_k|x_k), k \in \mathbb{N}\} \tag{47}$$

The observation process is encapsulated in a *single* pseudo-sensor producing a *single* measurement $z_k$ at each time step. Likewise, any false alarm, detection or measurement-to-data issue is encapsulated in the likelihood function $g$.

Denote by $x_{0:k}$ (resp. $z_{0:k}$) the sequence of target states (resp. measurements) up to a given time step $k$, also called the the target (resp. measurement) trajectory up to $k$. In its most general form, the filtering problem aims at estimating quantities such as:

- the posterior density $p(x_{0:k}|z_{0:k})$;
- expectations of integrable functions $I(f_k) = \int_{\mathcal{X}^k} f_k(x_{0:k})p(x_{0:k}|z_{0:k})dx_{0:k}$.

given the collection of measurement $z_{0:k}$, usually under the assumptions that, at each time step $k$, one can:

- sample from transition density $t(.|x_{k-1})$;
- evaluate transition density $t(x_k|x_{k-1})$ a posteriori;
- sample from likelihood $g(.|x_k)$;
- evaluate likelihood $g(z_k|x_k)$ a posteriori.

The Bayes rule provides another expression of the quantities to be estimated:

$$\begin{cases} p(x_{0:k}|z_{0:k}) \propto \tilde{p}(x_{0:k}) = p(z_{0:k}|x_{0:k})p(x_{0:k}) \\ I(f_k) = \dfrac{\int_{\mathcal{X}^k} f_k(x_{0:k})\tilde{p}(x_{0:k})dx_{0:k}}{\int_{\mathcal{X}^k} \tilde{p}(x_{0:k})dx_{0:k}} \end{cases} \tag{48}$$

The practical implementation of an estimator based on (48) may arise several well-known difficulties:

- one cannot sample from the posterior density $p(.|z_{0:k})$;
- the posterior density $p(.|z_{0:k})$ cannot be evaluated a posteriori;
- the integral in the expectation $I(f_k)$ cannot be evaluated.

## Importance sampling (IS)

The IS method is based on the practical assumption that, even if one cannot sample directly from the posterior density $p(.|z_{0:k})$, one can still design an *importance sampling density* $\pi(.|z_{0:k})$ which is "close enough" to the posterior density $p(x_{0:k}|z_{0:k})$ and yet easier to sample from. The principle of the method is based on the following theorem [Gewe 89]:

**Theorem 1.** *Under the following assumptions:*

1. $\tilde{p}(.)$ *is proportional to a proper density on* $\mathcal{X}^k$*;*
2. $\{x_{0:k}^{(i)}\}_{i=1}^{\infty}$ *is a collection of sequences of* $k$ *target states, i.i.d. according to* $\pi(.|z_{0:k})$*;*
3. *The support of* $\pi(.|z_{0:k})$ *includes* $\mathcal{X}^k$*;*
4. $I(f_k)$ *exists and is finite.*

*then:*

$$\hat{I}_N(f_k) = \frac{\sum_{i=1}^{N} f_k(x_{0:k}^{(i)})\tilde{p}(x_{0:k}^{(i)})/\pi(x_{0:k}^{(i)}|z_{0:k})}{\sum_{i=1}^{N} \tilde{p}(x_{0:k}^{(i)})/\pi(x_{0:k}^{(i)}|z_{0:k})} \xrightarrow{\mathcal{D}} I(f_k) \tag{49}$$

The theoretical assumptions in theorem 1 are quite mild, thus one can expect to approximate $I(f_k)$ properly provided that the *particle number* $N$ is large enough. In its algorithmic version, theorem 1 is often written with the more convenient *importance weights* notation:

---

**Algorithm 11** Importance Sampling (time $k$)

---

**input:** Measurement sequence up to current time: $z_{0:k}$
**output:** Weighted particles: $\{x_{0:k}^{(i)}, w_{0:k}^{(i)}\}_{i\in[1\ N]}$

Sampling
**for** $i = 1$ **to** $N$ **do**
  Sample target state sequence: $x_{0:k}^{(i)} \sim \pi(.|z_{0:k})$
  Compute weight: $\tilde{w}_{0:k}^{(i)} \propto \frac{p(z_{0:k}|x_{0:k}^{(i)})p(x_{0:k}^{(i)})}{\pi(x_{0:k}^{(i)}|z_{0:k})}$
**end for**

Normalization
**for** $i = 1$ **to** $N$ **do**
  Normalize weight: $w_{0:k}^{(i)} \leftarrow \frac{\tilde{w}_{0:k}^{(i)}}{\sum_{j=1}^{N} \tilde{w}_{0:k}^{(j)}}$
**end for**

Estimation
Approximate posterior distribution: $p(.|z_{0:k}) \simeq \sum_{i=1}^{N} w_{0:k}^{(i)} \delta_{x_{0:k}^{(i)}}(.)$
Approximate expectation: $I(f_k) \simeq \sum_{i=1}^{N} w_{0:k}^{(i)} f_k(x_{0:k}^{(i)})$

---

An important remark concerning algorithm 11 is that, besides the *theoretical* assumptions given in theorem 1, it requires the following *practical* assumptions:

- one can sample from the importance density $\pi(.|z_{0:k})$;
- one can evaluate the ratio $\frac{p(x_{0:k}|z_{0:k})}{\pi(x_{0:k}|z_{0:k})}$ a posteriori.

In other words, the importance sampling bypasses the sampling from the posterior density $p(.|z_{0:k})$ by considering an "nicer" importance density.

# Sequential Importance Sampling (SIS)

The IS method suffers from a major drawback that may hinder both its tractability and/or its computational efficency. By construction, the IS method (algorithm 11) is *not* recursive. At every time step $k$, newly drawn samples $x_{0:k}^{(i)}$ belongs to the "enlarged" state space $\mathcal{X}^k$ and are independent from the previous samples $x_{0:k-1}^{(i)}$ - conditionally on the sequence of measurements $z_{0:k}$. Likewise, the new weights $w_{0:k}^{(i)}$ are independent from the previous weights $w_{0:k-1}^{(i)}$. That is, one must "start from scratch" at each iteration. Besides, the design of the importance sampling is quite challenging without additional assumptions. Indeed, being able to sample from and evaluate a posteriori the *one-step* transition $t(.|x_{k-1})$ and the likelihood $g(.|x_k)$ seems hardly sufficient to be able to:

- draw a target trajectory from $\pi(.|z_{0:k})$;
- evaluate the probability of occurence of any $k$-step trajectory $x_{0:k}$ through $\pi(x_{0:k}|z_{0:k})$, let alone $p(x_{0:k}|z_{0:k})$.

## Design of the iterative method

The salient feature of the SIS method is to propose a *recursive* sampling and weighting of the target trajectories by considering importance densities such that [Douc 00]:

$$\pi(x_{0:k}|z_{0:k}) = q(x_0|z_0)\prod_{j=1}^{k} q(x_k|x_{0:k-1}, z_{0:k}) \tag{50}$$

where $q(.|x_{0:k-1}, z_{0:k})$ is the *importance function*. Thus, the importance weight $w_{0:k}^{(i)}$ can be built recursively as follows:

$$\begin{aligned} w_{0:k}^{(i)} &\propto \frac{p(z_{0:k}|x_{0:k}^{(i)})p(x_{0:k}^{(i)})}{\pi(x_{0:k}^{(i)}|z_{0:k})} \\ &\propto \frac{p(z_{0:k}|x_{0:k}^{(i)})p(x_{0:k}^{(i)})}{\pi(x_{0:k-1}^{(i)}|z_{0:k-1})q(x_k^{(i)}|x_{0:k-1}^{(i)}, z_{0:k})} \end{aligned}$$

$$
\begin{aligned}
w_{0:k}^{(i)} &\propto \frac{\overbrace{p(z_{0:k-1}|x_{0:k}^{(i)},z_k)}^{=p(z_{0:k-1}|x_{0:k-1}^{(i)})}\overbrace{p(z_k|x_{0:k}^{(i)})}^{=g(z_k|x_k^{(i)})}\overbrace{p(x_k^{(i)}|x_{0:k-1}^{(i)})}^{=t(x_k^{(i)}|x_{k-1}^{(i)})}p(x_{0:k-1}^{(i)})}{\pi(x_{0:k-1}^{(i)}|z_{0:k-1})q(x_k^{(i)}|x_{0:k-1}^{(i)},z_{0:k})} \\
&\propto \frac{p(z_{0:k-1}|x_{0:k-1}^{(i)})p(x_{0:k-1}^{(i)})}{\pi(x_{0:k-1}^{(i)}|z_{0:k-1})}\frac{g(z_k|x_k^{(i)})t(x_k^{(i)}|x_{k-1}^{(i)})}{q(x_k^{(i)}|x_{0:k-1}^{(i)},z_{0:k})} \\
&\propto w_{0:k-1}^{(i)}\frac{g(z_k|x_k^{(i)})t(x_k^{(i)}|x_{k-1}^{(i)})}{q(x_k^{(i)}|x_{0:k-1}^{(i)},z_{0:k})}
\end{aligned}
$$

**Algorithm 12** Sequential Importance Sampling (time $k$)

**input:** Measurement sequence up to current time: $z_{0:k}$
**output:** Previous weighted particles: $\{x_{0:k-1}^{(i)}, w_{0:k-1}^{(i)}\}_{i\in[1\ N]}$
**output:** Current weighted particles: $\{x_{0:k}^{(i)}, w_{0:k}^{(i)}\}_{i\in[1\ N]}$

Sampling
**for** $i = 1$ **to** $N$ **do**
  Sample new target state: $x_k^{(i)} \sim q(.|x_{k-1}^{(i)}, z_{0:k})$
  Update target sequence: $x_{0:k}^{(i)} \leftarrow (x_{0:k-1}^{(i)}, x_k^{(i)})$
  Compute weight: $\tilde{w}_k^{(i)} \propto w_{k-1}^{(i)}\frac{g(z_k|x_k^{(i)})t(x_k^{(i)}|x_{k-1}^{(i)})}{q(x_k^{(i)}|x_{0:k-1}^{(i)},z_{0:k})}$
**end for**

Normalization
**for** $i = 1$ **to** $N$ **do**
  Normalize weight: $w_k^{(i)} \leftarrow \frac{\tilde{w}_k^{(i)}}{\sum_{j=1}^N \tilde{w}_k^{(j)}}$
**end for**

Estimation
Approximate posterior distribution: $p(.|z_{0:k}) \simeq \sum_{i=1}^N w_k^{(i)}\delta_{x_{0:k}^{(i)}}(.)$
Approximate expectation: $I(f_k) \simeq \sum_{i=1}^N w_k^{(i)} f_k(x_{0:k}^{(i)})$

The practical assumptions of the SIS algorithm are the following:

- one can sample from the importance function $q(.|x_{k-1}^{(i)}, z_{0:k})$;
- one can evaluate the ratio $\frac{g(z_k|x_k^{(i)})t(x_k^{(i)}|x_{k-1}^{(i)})}{q(x_k^{(i)}|x_{0:k-1}^{(i)},z_{0:k})}$ a posteriori.

Clearly these assumptions are weaker that those required for the IS algorithm (see algorithm 11) as only one-step densities are involved, all the more since transition $t(.|x_{k-1}^{(i)})$ and likelihood $g(.|x_k^{(i)})$ densities are supposed to be easy to sample from and evaluate a posteriori. Understandably, the design of the importance function $q(.|x_{k-1}^{(i)}, z_{0:k})$ is critical to the quality of the approximation.

Note that the trajectory samples $x_{0:k}^{(i)}$ are no longer drawn "from scratch" as in the IS algorithm, but rather updated from the previous trajectory sample with the same label (i.e. the tail of $x_{0:k}^{(i)}$ is precisely $x_{0:k-1}^{(i)}$). Therefore, to the author's knowledge, it is unclear if the second assumption from the fundamental theorem (see theorem 1) still hold in this case, and thus if the convergence in distribution is still guaranteed.

## Choice of the importance function

An excessive variance in the distribution of the importance weights (known as "particle degeneracy") is often considered unsatisfying because it means that computing resources are likely to be wasted on the update of particles with negligible weight and thus negligible effect in the approximation (see algorithm 12). Therefore, the optimal importance function is defined as the one that minimizes the variance among the importance weights [Douc 00]:

**Theorem .1.** *The optimal importance function is defined as:*

$$q_{opt}(.|x_{0:k-1}^{(i)}, z_{0:k}) = \arg\min_q Var_q[\tilde{w}_k^{(i)}] \tag{51}$$

*and we have:*

$$q_{opt}(.|x_{0:k-1}^{(i)}, z_{0:k}) = p(.|x_{k-1}^{(i)}, z_k) \tag{52}$$

*Proof.*

$$\begin{aligned}
&Var_q[\tilde{w}_k^{(i)}] = \mathbb{E}_q[(\tilde{w}_k^{(i)})^2] - \mathbb{E}_q^2[\tilde{w}_k^{(i)}] \\
&= \int q(x|x_{0:k-1}^{(i)}, z_{0:k})(\tilde{w}_k^{(i)}(x))^2 dx - \left( \int q(x|x_{0:k-1}^{(i)}, z_{0:k}) \tilde{w}_k^{(i)}(x) dx \right)^2 \\
&= \int q(x|x_{0:k-1}^{(i)}, z_{0:k}) \left( w_{k-1}^{(i)} \frac{g(z_k|x)t(x|x_{k-1}^{(i)})}{q(x|x_{0:k-1}^{(i)}, z_{0:k})} \right)^2 dx \\
&\quad - \left( \int q(x|x_{0:k-1}^{(i)}, z_{0:k}) w_{k-1}^{(i)} \frac{g(z_k|x)t(x|x_{k-1}^{(i)})}{q(x|x_{0:k-1}^{(i)}, z_{0:k})} dx \right)^2 \\
&= (w_{k-1}^{(i)})^2 \left[ \int \frac{g^2(z_k|x)t^2(x|x_{k-1}^{(i)})}{q(x|x_{0:k-1}^{(i)}, z_{0:k})} dx - \left( \int g(z_k|x)t(x|x_{k-1}^{(i)}) dx \right)^2 \right]
\end{aligned}$$

Thus, with $q(x|x_{0:k-1}^{(i)}, z_{0:k}) = p(x|x_{k-1}^{(i)}, z_k) = \frac{g(z_k|x)t(x|x_{k-1}^{(i)})}{p(z_k|x_{k-1}^{(i)})}$:

$$
\begin{aligned}
& Var_q[\tilde{w}_k^{(i)}] \\
& = (w_{k-1}^{(i)})^2 \left[ \int \frac{g^2(x_k|x)t^2(x|x_{k-1}^{(i)})p(z_k|x_{k-1}^{(i)})}{g(z_k|x)t(x|x_{k-1}^{(i)})} dx - \left( \int g(z_k|x)t(x|x_{k-1}^{(i)})dx \right)^2 \right] \\
& = (w_{k-1}^{(i)})^2 \left[ p(z_k|x_{k-1}^{(i)}) \underbrace{\int g(z_k|x)t(x|x_{k-1}^{(i)})dx}_{=p(z_k|x_{k-1}^{(i)})} - \left( \underbrace{\int g(z_k|x)t(x|x_{k-1}^{(i)})dx}_{=p(z_k|x_{k-1}^{(i)})} \right)^2 \right] \\
& = 0
\end{aligned}
$$

□

With the optimal importance function, the importance weights update in algorithm 12 is simplified:

$$
\begin{aligned}
\tilde{w}_k^{(i)} &\propto w_{k-1}^{(i)} \frac{g(z_k|x_k^{(i)})t(x_k^{(i)}|x_{k-1}^{(i)})}{q_{opt}(x_k^{(i)}|x_{0:k-1}^{(i)}, z_{0:k})} \\
&\propto w_{k-1}^{(i)} \frac{g(z_k|x_k^{(i)})t(x_k^{(i)}|x_{k-1}^{(i)})p(z_k|x_{k-1}^{(i)})}{g(z_k|x_k^{(i)})t(x_k^{(i)}|x_{k-1}^{(i)})} \\
&\propto w_{k-1}^{(i)} p(z_k|x_{k-1}^{(i)})
\end{aligned} \tag{53}
$$

The optimal solution is often impractical because the optimal importance function (52) cannot be sampled from and/or the quantity $p(z_k|x_{k-1}^{(i)})$ in the weight update (53) cannot be evaluated a posteriori - recall from the general description that the only densities that can be easily handled are the one-step transition $t(.|x_{k-1})$ and likelihood $g(.|x_k)$. The simplest solution is to use the one-step transition as importance function, in this case the importance weights update in algorithm 12 is further simplified:

$$
\begin{aligned}
\tilde{w}_k^{(i)} &\propto w_{k-1}^{(i)} \frac{g(z_k|x_k^{(i)})t(x_k^{(i)}|x_{k-1}^{(i)})}{q_{opt}(x_k^{(i)}|x_{0:k-1}^{(i)}, z_{0:k})} \\
&\propto w_{k-1}^{(i)} \frac{g(z_k|x_k^{(i)})t(x_k^{(i)}|x_{k-1}^{(i)})}{t(x_k^{(i)}|x_{k-1}^{(i)})} \\
&\propto w_{k-1}^{(i)} g(z_k|x_k^{(i)})
\end{aligned} \tag{54}
$$

Using the one-step transition as importance function requires being able to:

- sample from transition density $t(.|x_{k-1})$;
- sample from likelihood $g(.|x_k)$;
- evaluate likelihood $g(z_k|x_k)$ a posteriori.

The key aspect of this solution is that it does not need to evaluate the transition density a posteriori but only to sample from it, which makes this solution tractable in almost every situation. By construction, one must be able to sample from the transition density to simulate the evolution of true targets; using the same method for the evolution of the particles is usually possible provided that the computational cost of the transition and/or the partition number is not too large. The evaluation a posteriori, however, is usually much more difficult. The is rather clear for the free target model (algorithm 1), and even clearer for the ground-based target model (algorithm 1 and figure 4.2). The assumptions on the likelihood function are not so restrictive in the single-sensor case. Indeed, the noise on the measurement process is typically Gaussian with zero mean and known variance, thus it is quite easy to sample from $g(.|x_k)$ and to evaluate $g(z_k|x_k)$ a posteriori (recall the closed-form expression of the likelihood $L_k^{j,\cdot}(.)$ in section 4.1.2).

**Filtrage PHD multicapteur avec application à la gestion de capteur**

**Résumé**
Le filtrage multiobjet est une technique de résolution du problème de détection et/ou suivi dans un contexte multicible. Cette thèse s'intéresse au filtre PHD (Probability Hypothesis Density), une célèbre approximation du filtre RFS (Random Finite Set) adaptée au cas où les observations sont le fruit d'un seul capteur. La première partie propose une construction rigoureuse du filtre PHD multicapteur exact et son expression simplifiée, sans approximation, grâce à un partitionnement joint de l'espace d'état des cibles et des capteurs. Avec cette nouvelle méthode, la solution exacte du filtre PHD multicapteur peut être propagée dans des scénarios de surveillance simples. La deuxième partie aborde le problème de gestion des capteurs dans le cadre du PHD. A chaque itération, le BET (Balanced Explorer and Tracker) construit une prédiction du PHD multicapteur a posteriori grâce au PIMS (Predicted Ideal Measurement Set) et définit un contrôle multicapteur en respectant quelques critères opérationnels simples adaptés aux missions de surveillance.

**Mots-clés : filtrage multiobjet, PHD multicapteur, gestion de capteurs**

---

**Multi-sensor PHD filtering with application to sensor management**

**Abstract**
The aim of multi-object filtering is to address the multiple target detection and/or tracking problem. This thesis focuses on the Probability Hypothesis Density (PHD) filter, a well-known tractable approximation of the Random Finite Set (RFS) filter when the observation process is realized by a single sensor. The first part proposes the rigorous construction of the exact multi-sensor PHD filter and its simplified expression, without approximation, through a joint partitioning of the target state space and the sensors. With this new method, the exact multi-sensor PHD can be propagated in simple surveillance scenarii. The second part deals with the sensor management problem in the PHD framework. At each iteration, the Balanced Explorer and Tracker (BET) builds a prediction of the posterior multi-sensor PHD thanks to the Predicted Ideal Measurement Set (PIMS) and produces a multi-sensor control according to a few simple operational principles adapted to surveillance activities.

**Keywords: multi-object filtering, multi-sensor PHD, sensor management**