

Approche variationnelle à noyau pour le suivi de cibles dans un réseau de capteurs sans fil.

Hichem Snoussi⁽¹⁾, Paul Honeine⁽¹⁾ et Cédric Richard⁽²⁾

⁽¹⁾ ICD (UMR STMR CNRS 6279), Université de Technologie de Troyes
12, rue Marie Curie, 10010, France
Email: snoussi@utt.fr

⁽²⁾ Laboratoire FIZEAU (UMR CNRS 6525), Université de Nice Sophia-Antipolis
Parc de Valrose, 06108 Nice, France
Email: cedric.richard@unice.fr

Résumé

Ce chapitre traite le problème de localisation décentralisée de cibles mobiles dans un réseau de capteurs sans fil, s'accommodant des contraintes d'énergie et de puissance limitées des capteurs embarqués. Nous décrivons une technique de localisation bénéficiant à la fois de la consistance de l'approche bayésienne et de la robustesse des méthodes à noyau. Cette technique repose sur un filtrage variationnel en ligne intégrant une phase d'apprentissage de la fonction de vraisemblance. Cette phase d'apprentissage rend cette méthode de localisation particulièrement robuste et flexible dans un environnement inconnu et non stationnaire.

0.1 Introduction

Les réseaux de capteurs provoquent un intérêt croissant au sein des communautés scientifiques et industrielles depuis plusieurs années [1]. Un réseau de capteurs intelligents peut être défini comme un système intégré comprenant, d'une part les moyens de réaliser une mesure, d'autre part les moyens de traiter cette mesure et diffuser l'information jusqu'à un système de niveau supérieur ou un humain. Les capteurs sont dotés de technologies sans fil et sont à même de communiquer de proche en proche, d'une façon dynamique et instantanée, reconfigurables en fonction de l'évolution de la population de capteurs, sans hiérarchie centrale, et de former ainsi une structure en forme de filet. Ce mode distribué présente l'avantage d'être particulièrement robuste aux attaques extérieures et à la défaillance des capteurs puisqu'il est prévu que la perte de composants ne compromette pas l'efficacité du réseau dans son ensemble. Aussi la formulation des problèmes de décision et des algorithmes de résolution s'en trouve-t-elle totalement modifiée puisque chaque noeud du réseau, composé d'un capteur intelligent miniaturisé et ayant une capacité énergétique limitée, se voit confié la mission de réaliser des mesures et de prendre des décisions locales partielles. Ces dernières sont affinées de proche en proche par les capteurs qui l'entourent. À l'heure actuelle, ces nouvelles technologies se retrouvent dans un grand nombre d'applications qui ont vocation à devenir de plus en plus nombreuses au fur et à mesure que les technologies d'intégration (micro et nanosystèmes) se développent. En effet, le déploiement rapide, le coût réduit et la tolérance aux pannes des réseaux de capteurs sont des caractéristiques qui les rendent un outil appréciable dans plusieurs domaines d'applications, tels que les activités militaires, les télécommunications, le domaine biomédical, etc (voir figure 1).

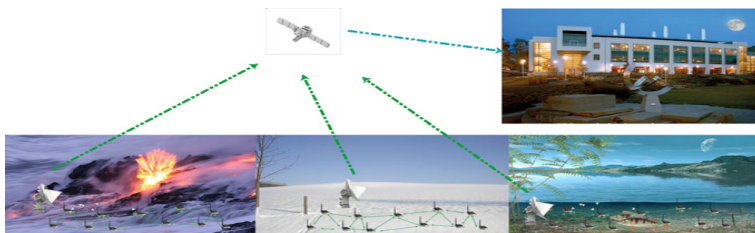


FIGURE 1 – Applications potentielles des réseaux de capteurs sans fil.

Les fonctions assurées par les réseaux de capteurs sans fil doivent s'accommoder des contraintes des communications numériques. En fait, dans ces systèmes, le traitement des données repose sur le caractère distribué et

coopératif des capteurs constituant le réseau pour une prise de décision plus sûre. Les noeuds composant le réseau sont autonomes et disposent pour cela d'une réserve énergétique dont le renouvellement peut s'avérer impossible, ce qui limite leur durée de vie. Chacun des noeuds doit être en mesure de traiter les données reçues, de prendre une décision locale et de la communiquer de façon autonome aux noeuds voisins auxquels il est connecté. Cette coopération est destinée à assurer les meilleures prises de décision possibles malgré les limites en termes de consommation énergétique et de puissance de traitement. Dans ce contexte, il est primordial que les solutions proposées soient coopératives et fassent appel à des techniques distribuées intelligentes, tant au niveau du mode de communication qu'à celui du traitement des informations acquises.

0.2 Etat de l'art : limitations des méthodes existantes

Le problème de suivi d'un objet mobile (cible passive ou capteur actif) est généralement résolu dans un cadre bayésien en se basant sur un modèle d'état. Le modèle d'état contient deux équations : une équation reflétant l'*a priori* qu'on possède sur la trajectoire de la cible et une deuxième équation reliant l'état inconnu du système aux observations dont disposent les capteurs réalisant le tracking. Le filtrage bayésien consiste à estimer la distribution *a posteriori* de l'état \mathbf{x}_t du système (position de la cible) sachant toutes les observations des capteurs ayant détecté la cible. Dans le cadre de ce travail, on considère le filtrage distribué (sans hiérarchie centrale) où seulement quelques capteurs, considérés comme pertinents, sont activés pour assurer un suivi de cibles efficace.

Des travaux récents ont été consacrés à l'implémentation du filtrage bayésien dans un réseau de capteurs sans fil. Considérant la non linéarité des équations d'observations, le filtrage bayésien est implémenté par une méthode de Monte Carlo séquentielle (filtrage particulaire) [17]. La popularité de ce type de méthodes est essentiellement due à leur flexibilité de traiter des modèles dynamiques non-linéaires/non-gaussiens et aussi des fonctions de vraisemblance (équations d'observation) non-linéaires/non-gaussiennes. Cependant, le filtrage particulaire nécessite l'échange d'un nombre important de particules (positions simulées aléatoirement) lorsque les capteurs assurant le tracking passent la main à d'autres capteurs plus pertinents à l'instant suivant. Pour cette raison, des approximations ont été considérées pour s'accommoder aux contraintes d'énergie. Dans [3], une approximation de la

distribution des particules de type KD-tree a été considérée. Dans [4], une stratégie collaborative basée sur une approximation de type mélange de gaussiennes et implémentée par un algorithme EM (Expectation-Maximization) a été proposée. Contrairement à l'approximation KD-tree, l'approximation par mélange de gaussiennes ne permet pas un contrôle de la propagation de l'erreur d'approximation. L'approche KD-tree permet d'avoir un compromis entre l'erreur d'approximation et les contraintes de communications numériques. Cependant, les deux approches induisent une propagation d'erreurs d'approximations successives. Récemment, une approche variationnelle collaborative a été proposée dans [2]. Elle consiste à mettre à jour en ligne une forme approchée de la distribution *a posteriori*. Cette approche permet une compression adaptative de la distribution non-gaussienne de l'état du système. La communication entre les capteurs se fait donc sans perte puisque l'échange se base seulement sur l'envoi d'une seule gaussienne. Le principal défaut de toutes les approches de tracking présentées ci-dessus est le fait qu'elles ne se soucient pas d'un facteur bien plus important dans les réseaux de capteurs sans fil : **la modalité d'observation**. Les points suivants doivent être pris en compte :

1. Pour des raisons de coût, les capteurs ne sont pas en général équipés de systèmes de capture sophistiqués.
2. Les conditions de fonctionnement réelles des capteurs ne sont pas en général connues. Il s'en suit que la fonction d'observation n'est pas connue et change avec l'environnement du capteur.
3. Les capteurs peuvent se dégrader avec le temps (usure normale ou incidents).

A titre d'exemple, le modèle de proximité qui est très utilisé dans la littérature est celui du RSSI (Received Signal Strength Indicator) [5]. Le RSSI est basé sur un modèle paramétrique dont les paramètres doivent être appris selon l'environnement du capteur. Les performances du tracking et de la localisation sont très sensibles à la pertinence de ce modèle et à ses paramètres fixés. En fait, le RSSI représente la forme de la vraisemblance qui est utilisée dans le filtrage bayésien. La déviation de cette fonction de vraisemblance cause de sévères dégradations des performances du tracking. Une manière d'éviter ce problème est l'utilisation des capteurs binaires [6] , [7], [8]. Un capteur binaire est basé sur la comparaison du RSSI par rapport à un seuil fixé pour décider si la cible est dans le voisinage ou non du capteur. Le principal défaut de cette approche est la perte d'une quantité importante de l'information en seuillant le RSSI (quantification binaire).

0.3 Suivi de cibles sans modèle

Pour résoudre le problème de localisation de cibles, on se place dans un cadre bayésien. Si $\mathbf{x}(t)$ est la position de la cible à estimer à l'instant t et $y_{t,m}$ est l'observation relevée par le capteur m à l'instant t , le modèle d'état s'écrit :

$$\begin{cases} \mathbf{x}_t & \sim p(\mathbf{x}_t | \mathbf{x}_{t-1}) \\ y_{t,m} & = g_m(\mathbf{x}_t) + \epsilon_t \end{cases}$$

où la fonction $g_m(\cdot)$, reliant la position de la cible à l'observation du capteur m , modélise le fonctionnement du capteur. Cette fonction est supposée en général connue dans les méthodes de tracking. Cependant, on peut noter qu'en pratique, la fonction g ne peut pas être parfaitement connue. De plus, cette fonction, même lorsqu'elle est valable, possède des paramètres qui varient en fonction de l'endroit géographique où se trouvent les capteurs. La figure 2 montre, à titre d'exemple, la puissance RSSI (Received Signal Strength) échangée entre 2 capteurs Mica2 (commercialisés par Crossbow) en fonction de leur distance. On peut noter l'imprécision de ce modèle qui est très utilisé dans la littérature. De plus, une simple rotation des capteurs, peut modifier complètement les paramètres de la courbe RSSI.

Nous proposons, dans ce travail, une méthodologie de tracking sans aucune information sur le modèle d'observation. La relation entre les données observées et la position de la cible est apprise d'une manière locale à partir seulement des données mesurées sur les capteurs. Le principe de la méthode consiste à exploiter les puissances échangées entre les capteurs eux-mêmes en plus des données issues de l'observation de la cible. En effet, comme les positions des capteurs sont connues, les données supplémentaires des puissances échangées permettent d'obtenir ce qu'on appelle dans la communauté "machine learning" des données d'apprentissage [11] (voir figure 3). Le problème de tracking est reformulé comme un problème de complétion matricielle. La résolution de cette complétion matricielle aboutit à une relation linéaire/gaussienne entre les observations et l'état inconnu du système (position géographique). La linéarité de la relation obtenue permet de résoudre le problème de tracking d'une manière très efficace et rapide de point de vue implémentation. En effet, deux algorithmes peuvent être utilisés : un filtrage de Kalman ou un filtrage variationnel. Par rapport à son implémentation classique [2], le filtrage variationnel, grâce à la linéarité de la relation entre les observations et l'état du système, devient très rapide.

Les avantages de cette méthode par rapport aux approches classiques peuvent se résumer dans les points suivants :

1. Tracking sans modèle d'observation grâce à un apprentissage local de la relation entre la position de la cible et les observations.
2. Robustesse par rapport à un environnement non stationnaire (voir figure 4).
3. Robustesse par rapport à une déviation d'un modèle même s'il est connu (vieillesse ou dégradation du capteur,...).
4. Implémentation rapide du filtrage variationnel grâce à la linéarité des équations obtenus par la complétion matricielle.

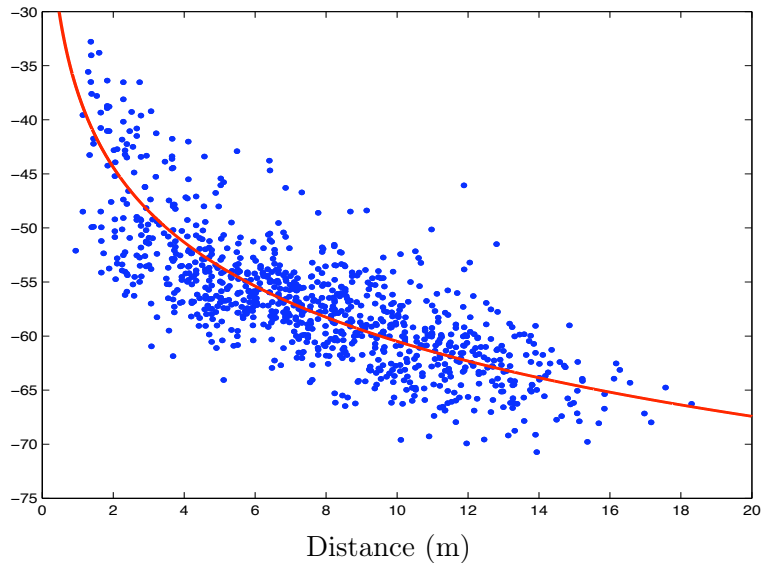


FIGURE 2 – Imprécision du modèle reliant la distance à la puissance échangée entre deux capteurs Mica2 (produit Crossbow <http://www.xbow.com/>).

0.3.1 Construction de la vraisemblance par régression matricielle

Cette sous-section est consacrée à expliquer les aspects techniques de la construction locale d'un modèle de vraisemblance linéaire et gaussien en exploitant les données mesurées entre des capteurs ayant des positions connues. Afin d'établir un lien avec le problème de régression matricielle, les données observées sont considérées comme des données de similarité entre le capteur et la cible. Notre unique hypothèse de départ sur la nature des observations est le fait que les données de similarité mesurées entre les capteurs et les cibles

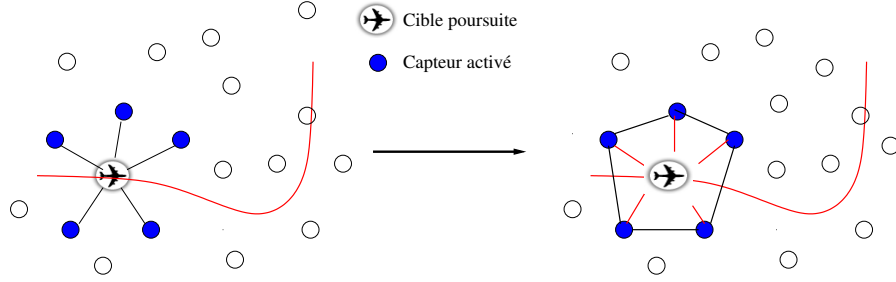


FIGURE 3 – Les capteurs (avec des positions connues) peuvent mesurer leurs propres RSSI \implies Données d'apprentissage !

ou entre les capteurs eux-mêmes est une fonction strictement décroissante de la distance. Aucun modèle n'est supposé connu.

A chaque instant t , on suppose qu'un ensemble de n capteurs $\{\mathbf{s}_1^{(t)}, \dots, \mathbf{s}_n^{(t)}\}$ sont sélectionnés pour estimer la position de la cible \mathbf{x}_t . On suppose de plus que les données de similarité mesurées entre les capteurs et la cible et entre les capteurs eux-mêmes sont disponibles au niveau d'un capteur sélectionné pour la mise à jour du processus de filtrage (voir sous-section 0.3.2). Les données de similarité inter-capteurs jouent le rôle de données d'apprentissage qui vont être exploitées sans connaissance du modèle qui les lie aux distances (voir figure 3). En suivant le principe du "kernel trick", communément utilisé dans la communauté machine learning, les données de similarité sont considérées comme des produits scalaires dans l'espace de Hilbert à noyau reproduisant (RKHS). En d'autres termes, la mesure de similarité entre un capteur $\mathbf{s}_i^{(t)}$ et un autre capteur $\mathbf{s}_j^{(t)}$ est considérée comme le produit scalaire euclidien entre leurs représentants $\phi(\mathbf{s}_i^{(t)})$ et $\phi(\mathbf{s}_j^{(t)})$ dans le RKHS : $\mathcal{K}(\mathbf{s}_i^{(t)}, \mathbf{s}_j^{(t)}) = \langle \phi(\mathbf{s}_i^{(t)}), \phi(\mathbf{s}_j^{(t)}) \rangle$.

Selon cette formulation, la matrice $(N \times N)$ -matrix (avec $N = n + 1$) des données de similarité correspond à la matrice noyau \mathbf{K} (mesurée et donc parfaitement connue) dont les éléments sont définis comme suit :

$$\begin{cases} (K)_{i,j} = \mathcal{K}(\mathbf{s}_i^{(t)}, \mathbf{s}_j^{(t)}) & 1 \leq i \neq j \leq n, \\ (K)_{i,n+1} = \mathcal{K}(\mathbf{s}_i^{(t)}, \mathbf{x}_t) & 1 \leq i \leq n, \\ (K)_{l,l} = c = \text{const.} & 1 \leq l \leq n + 1. \end{cases}$$

Comme la position de la cible est inconnue, la matrice \mathbf{G} ($N \times N$) formée

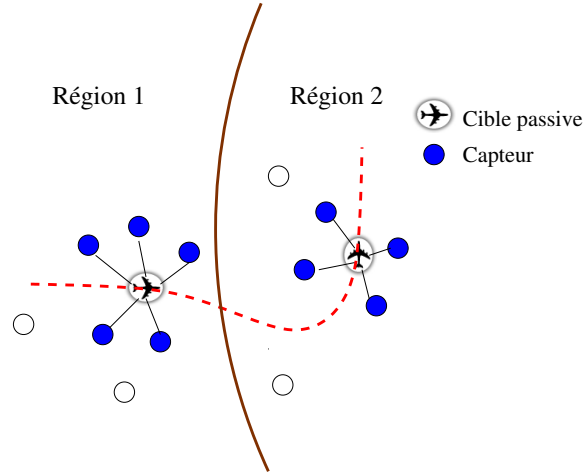


FIGURE 4 – Robustesse du tracking dans des environnements non stationnaires : les paramètres du modèle d’observation peuvent varier en fonction de la région où les capteurs sont déployés.

par les produits scalaires euclidiens de l’ensemble $\{\mathbf{s}_1^{(t)}, \mathbf{s}_2^{(t)}, \dots, \mathbf{s}_n^{(t)}, \mathbf{x}_t\}$ possède des entrées inconnues correspondant aux produits scalaires entre les capteurs de positions connues et la cible de position inconnue. L’objectif de la complétion matricielle consiste à estimer les entrées manquantes de la matrice \mathbf{G} en exploitant une forme de corrélation avec la matrice complète \mathbf{K} . En partageant la matrice \mathbf{G} en 4 blocks \mathbf{G}_{tt} , \mathbf{G}_{tp} , \mathbf{G}_{pt} and \mathbf{G}_{pp} correspondant respectivement aux produits scalaires capteur/capteur, capteur/cible, cible/capteur et cible/cible, le problème de complétion matricielle peut être illustré par le diagramme suivant :

$$\begin{array}{|c|c|} \hline \mathbf{K}_{tt} & \mathbf{K}_{tp} \\ \hline \mathbf{K}_{pt} & \mathbf{K}_{pp} \\ \hline \end{array} \underbrace{\hspace{10em}}_{\mathbf{K}} \rightarrow \begin{array}{|c|c|} \hline \mathbf{G}_{tt} & \mathbf{G}_{tp} \\ \hline \mathbf{G}_{pt} & \mathbf{G}_{pp} \\ \hline \end{array} \underbrace{\hspace{10em}}_{\mathbf{G}} \quad (1)$$

où l’objectif est la prédiction des blocks inconnus (en gris) \mathbf{G}_{tp} , \mathbf{G}_{pt} et \mathbf{G}_{pp} en estimant une relation entre les blocks (connus) \mathbf{K}_{tt} et \mathbf{G}_{tt} . Notons que dans notre problème de tracking seul le block \mathbf{G}_{tp} nous intéresse puisqu’il

contient les produits scalaires entre les capteurs et la cible :

$$\mathbf{G}_{tp} = [(\mathbf{s}_1^{(t)})^T \mathbf{x}_t, \dots, (\mathbf{s}_i^{(t)})^T \mathbf{x}_t, \dots, (\mathbf{s}_n^{(t)})^T \mathbf{x}_t]^T, \quad (2)$$

Il est utile de noter que la matrice \mathbf{G}_{tp} est linéaire par rapport à la position inconnue de la cible \mathbf{x}_t . Cette propriété sera exploitée pour une implémentation efficace du filtre variationnel dans la sous-section 0.3.2.

Afin de résoudre le problème de complétion matricielle, une méthode de régression matricielle a été proposée dans [12]. Elle se base essentiellement sur la formulation du problème dans l'espace RKHS. Dans cette invention, au lieu de prédire le block manquant \mathbf{G}_{tp} , on propose d'étendre cette méthode pour calculer une distribution de probabilité de \mathbf{G}_{tp} . En utilisant le même raisonnement de [12], on peut montrer que la relation entre les blocks de la matrice complète \mathbf{K} et les blocks de la matrice incomplète \mathbf{G} s'écrit comme suit :

$$\mathbf{G}_{tt} = \mathbf{K}_{tt} \mathbf{A} \mathbf{K}_{tt} + \mathbf{\Psi}_{tt} \quad (3)$$

$$\mathbf{G}_{tp} = \mathbf{K}_{tt} \mathbf{A} \mathbf{K}_{tp} + \mathbf{\Psi}_{tp} \quad (4)$$

$$\mathbf{G}_{pp} = \mathbf{K}_{pt} \mathbf{A} \mathbf{K}_{tp} + \mathbf{\Psi}_{pp} \quad (5)$$

où \mathbf{A} est une matrice inconnue et $\mathbf{\Psi} = (\epsilon_{i,j})_{i,j=1..N}$ est une matrice $(N \times N)$ de variables gaussiennes (i.i.d) avec la même variance $\sigma_{ij}^2 = \sigma^2$.

Selon la formulation statistique ci-dessus du problème de régression, on peut montrer que, connaissant les matrices \mathbf{G}_{tt} , \mathbf{K}_{tt} et \mathbf{K}_{tp} , la matrice \mathbf{G}_{tp} est aléatoire gaussienne dont la moyenne et la covariance sont données par les expressions suivantes :

$$\begin{cases} \boldsymbol{\mu}_g &= \mathbf{G}_{tt} \mathbf{K}_{tt}^{-1} \mathbf{K}_{tp} \\ \boldsymbol{\Sigma}_g &= \sigma^2 (\mathbf{K}_{pt} \mathbf{K}_{tt}^{-2} \mathbf{K}_{tp} + \mathbf{1}) \mathbf{I}_n \end{cases} \quad (6)$$

où \mathbf{I}_n la matrice $(n \times n)$ identité.

La distribution gaussienne du vecteur \mathbf{G}_{tp} est l'un des points clés de cette invention. En fait, en notant $\mathbf{S} = [\mathbf{s}_1^{(t)}, \mathbf{s}_2^{(t)}, \dots, \mathbf{s}_n^{(t)}]^T$ la matrice $(n \times 2)$ des positions des capteurs sélectionnés à l'instant t , l'aspect gaussien de \mathbf{G}_{tp} peut être exprimé par la relation suivante :

$$\mathbf{G}_{tp} = \mathbf{S} \mathbf{x}_t = \mathbf{G}_{tt} \mathbf{K}_{tt}^{-1} \mathbf{K}_{tp} + \boldsymbol{\gamma}_t \quad (7)$$

où $\boldsymbol{\gamma}_t$ est un bruit gaussien de moyenne nulle et de matrice de covariance diagonale $\boldsymbol{\Sigma}_g$ définie par (6). L'expression (7) peut être considérée comme le modèle statistique résultant liant les données observées et les positions

des objets mobiles (cibles passives ou capteurs mobiles). Ce modèle jouera le rôle de la fonction de vraisemblance lorsqu'on va implémenter le filtrage bayésien dans la prochaine sous-section 0.3.2. La quantité $\mathbf{G}_{tt}\mathbf{K}_{tt}^{-1}\mathbf{K}_{tp}$ dans le terme à droite de l'équation (7) peut être interprétée comme la statistique suffisante résumant toutes les données disponibles à l'instant courant t .

0.3.2 Filtrage variationnel pour le tracking des objets mobiles

Dans la suite, la fonction de vraisemblance sera basée sur le modèle linéaire gaussien (7) obtenu par la méthode de régression matricielle décrite dans la sous-section précédente. La dynamique de l'état du système \mathbf{x}_t est décrite par un modèle de mélange continu de gaussiennes (mean-scale mixture). Selon ce modèle, introduit dans [15] pour le suivi visuel de cible, l'état caché $\mathbf{x}_t \in \mathbb{R}^{n_x}$ suit une distribution gaussienne de moyenne aléatoire $\boldsymbol{\mu}_t$ et de matrice de précision aléatoire $\boldsymbol{\lambda}_t$. La moyenne suit une marche aléatoire gaussienne, traduisant la corrélation temporelle de la trajectoire de l'état caché du système et la matrice de précision suit une loi de Wishart :

$$\begin{cases} \boldsymbol{\mu}_t \sim \mathcal{N}(\boldsymbol{\mu}_t \mid \boldsymbol{\mu}_{t-1}, \bar{\boldsymbol{\lambda}}) \\ \boldsymbol{\lambda}_t \sim \mathcal{W}_{\bar{n}}(\boldsymbol{\lambda}_t \mid \bar{\mathbf{S}}) \\ \mathbf{x}_t \sim \mathcal{N}(\mathbf{x}_t \mid \boldsymbol{\mu}_t, \boldsymbol{\lambda}_t) \end{cases} \quad (8)$$

où les hyperparamètres $\bar{\boldsymbol{\lambda}}$, \bar{n} et $\bar{\mathbf{S}}$ sont respectivement la matrice de précision de la marche aléatoire, le degré de liberté et la matrice de précision de la distribution de Wishart. Il convient de noter que l'aspect aléatoire de la moyenne et de la précision induit une distribution *a priori* marginale dont le comportement des queues peut être ajusté d'une manière simple selon les valeurs des hyperparamètres [14]. De plus, une distribution à queues lourdes permet un suivi efficace de trajectoires présentant des sauts brusques. En effet, la distribution marginale de transition est obtenue en intégrant par rapport à la moyenne et à la matrice de précision aléatoires :

$$p(\mathbf{x}_t \mid \mathbf{x}_{t-1}) = \int \mathcal{N}(\mathbf{x}_t \mid \boldsymbol{\mu}_t, \boldsymbol{\lambda}_t) p(\boldsymbol{\mu}_t, \boldsymbol{\lambda}_t \mid \mathbf{x}_{t-1}) d\boldsymbol{\mu}_t d\boldsymbol{\lambda}_t \quad (9)$$

où l'intégration par rapport à la matrice de précision fait apparaître la classe connue des distributions hyperboliques généralisées (ou "scale-mixtures") introduite par Barndorff-Nielsen [14]. La valeur du degré de liberté \bar{n} de la distribution de Wishart conditionne fortement le comportement des queues

de la distribution marginale (9). La figure 5 montre des exemples de distributions hyperboliques généralisées scalaires obtenues en variant ses paramètres. On peut noter la flexibilité de cette famille de lois pour couvrir des comportements très variés des queues des distributions de probabilité. Il convient aussi de noter que, malgré la linéarité de la fonction de vraisemblance, l'application du filtre de Kalman est impossible à cause de la non linéarité de la dynamique de transition (8).

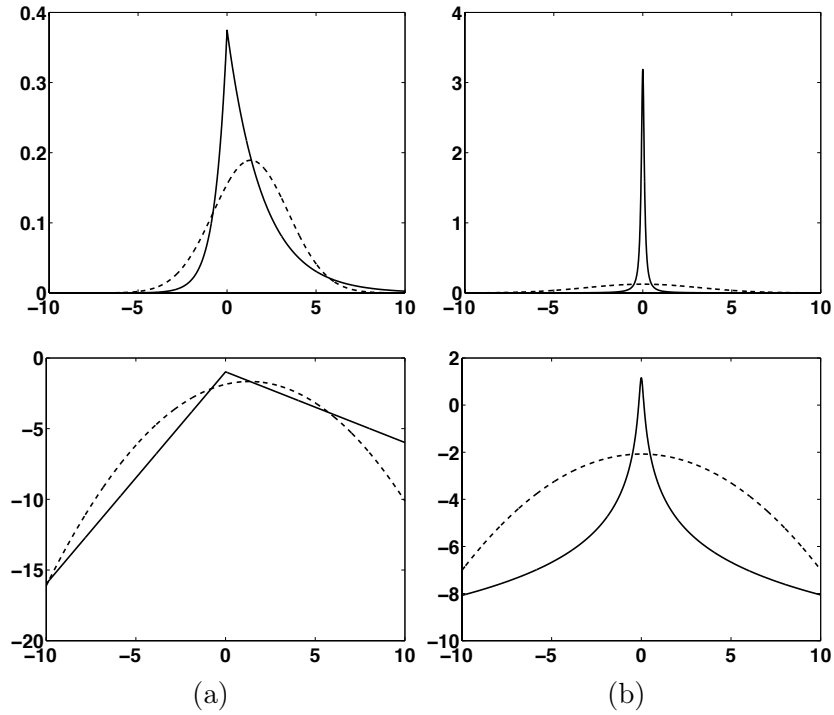


FIGURE 5 – Exemples de distributions hyperboliques généralisées :

(a) cas hyperbolique ;

(b) cas Cauchy.

Les Pdfs en première ligne et les log-densités en deuxième ligne. La ligne en pointillé correspond à la distribution gaussienne ayant la même moyenne et la même variance.

Selon le modèle de transition (8), l'état caché augmenté devient $\alpha_t = (\mathbf{x}_t, \boldsymbol{\mu}_t, \boldsymbol{\lambda}_t)$. Au lieu d'approcher la distribution de filtrage $p(\alpha_t | \mathbf{y}_{1..t})$ par un ensemble de particules pondérées (filtrage particulaire [16], [6]), le principe de l'approche variationnelle en ligne consiste à approcher cette distribution

par une autre fonctionnelle plus simple $q(\boldsymbol{\alpha}_t)$ en minimisant la divergence de Kullback-Leibler par rapport à la vraie distribution de filtrage :

$$D_{\text{KL}}(q||p) = \int q(\boldsymbol{\alpha}_t) \log \frac{q(\boldsymbol{\alpha}_t)}{p(\boldsymbol{\alpha}_t | \mathbf{y}_{1..t})} d\boldsymbol{\alpha}_t \quad (10)$$

En imposant une forme séparable (non paramétrique) $q(\boldsymbol{\alpha}_t) = q(\mathbf{x}_t)q(\boldsymbol{\mu}_t)q(\boldsymbol{\lambda}_t)$ et en minimisant la divergence de Kullback-Leibler (10) avec les outils du calcul variationnel, on obtient la procédure itérative suivante :

$$\begin{cases} q(\mathbf{x}_t) \propto \exp \langle \log p(\mathbf{y}_{1..t}, \boldsymbol{\alpha}_t) \rangle_{q(\boldsymbol{\mu}_t)q(\boldsymbol{\lambda}_t)} \propto \mathcal{N}(\mathbf{x}_t | \mathbf{x}_t^*, \boldsymbol{\Gamma}_t^*) \\ q(\boldsymbol{\mu}_t) \propto \exp \langle \log p(\mathbf{y}_{1..t}, \boldsymbol{\alpha}_t) \rangle_{q(\mathbf{x}_t)q(\boldsymbol{\lambda}_t)} \propto \mathcal{N}(\boldsymbol{\mu}_t | \boldsymbol{\mu}_t^*, \boldsymbol{\lambda}_t^*) \\ q(\boldsymbol{\lambda}_t) \propto \exp \langle \log p(\mathbf{y}_{1..t}, \boldsymbol{\alpha}_t) \rangle_{q(\mathbf{x}_t)q(\boldsymbol{\mu}_t)} \propto \mathcal{W}_{n^*}(\boldsymbol{\lambda}_t | \mathbf{S}_t^*) \end{cases} \quad (11)$$

où les paramètres sont mis à jour d'une manière itérative selon le schéma suivant :

$$\begin{aligned} \mathbf{x}_t^* &= \boldsymbol{\Gamma}_t^{*-1} (\mathbf{S}^T \boldsymbol{\Sigma}_g^{-1} \mathbf{G}_{tt} \mathbf{K}_{tt}^{-1} \mathbf{K}_{tp} + \langle \boldsymbol{\lambda}_t \rangle \langle \boldsymbol{\mu}_t \rangle) \\ \boldsymbol{\Gamma}_t^* &= \mathbf{S}^T \boldsymbol{\Sigma}_g^{-1} \mathbf{S} + \langle \boldsymbol{\lambda}_t \rangle \\ \boldsymbol{\mu}_t^* &= \boldsymbol{\lambda}_t^{*-1} (\langle \boldsymbol{\lambda}_t \rangle \langle \mathbf{x}_t \rangle + \boldsymbol{\lambda}_t^p \boldsymbol{\mu}_t^p) \\ \boldsymbol{\lambda}_t^* &= \langle \boldsymbol{\lambda}_t \rangle + \boldsymbol{\lambda}_t^p \\ n^* &= \bar{n} + 1 \\ \mathbf{S}_t^* &= (\langle \mathbf{x}_t \mathbf{x}_t^T \rangle - \langle \mathbf{x}_t \rangle \langle \boldsymbol{\mu}_t \rangle^T - \langle \boldsymbol{\mu}_t \rangle \langle \mathbf{x}_t \rangle^T + \langle \boldsymbol{\mu}_t \boldsymbol{\mu}_t^T \rangle + \bar{\mathbf{S}}^{-1})^{-1} \\ \boldsymbol{\mu}_t^p &= \boldsymbol{\mu}_{t-1}^* \\ \boldsymbol{\lambda}_t^p &= (\boldsymbol{\lambda}_{t-1}^{*-1} + \bar{\boldsymbol{\lambda}}^{-1})^{-1} \end{aligned} \quad (12)$$

Notons que toutes les variables ont des espérances mathématiques simples à calculer :

$$\begin{cases} \langle \mathbf{x}_t \rangle = \mathbf{x}_t^*, & \langle \mathbf{x}_t \mathbf{x}_t^T \rangle = \boldsymbol{\Gamma}_t^{*-1} + \mathbf{x}_t^* \mathbf{x}_t^{*T}, \\ \langle \boldsymbol{\mu}_t \rangle = \boldsymbol{\mu}_t^*, & \langle \boldsymbol{\mu}_t \boldsymbol{\mu}_t^T \rangle = \boldsymbol{\lambda}_t^{*-1} + \boldsymbol{\mu}_t^* \boldsymbol{\mu}_t^{*T}, \\ \langle \boldsymbol{\lambda}_t \rangle = n^* \mathbf{S}_t^* \end{cases}$$

Remarque 1 Dans les travaux précédents [15], [2], pour une fonction de vraisemblance de forme générale $p(\mathbf{y}_t | \mathbf{x}_t)$, la position de la cible ne possède pas une distribution $q(\mathbf{x}_t)$ analytique simple. Pour calculer sa moyenne et sa covariance, il a fallu procéder par des simulations de Monte-Carlo. Cependant, dans ce travail, grâce à la forme linéaire et gaussienne de la vraisemblance, la moyenne et la covariance de \mathbf{x}_t sont calculées d'une manière exacte, d'où une implémentation rapide de l'algorithme variationnel.

Remarque 2 Il convient de noter que le calcul de $q(\boldsymbol{\alpha}_t)$ est implémenté d'une manière séquentielle (dans le temps) en se basant uniquement sur la connaissance de $q(\boldsymbol{\mu}_{t-1})$. En effet, en tenant en compte la forme séparable de la distribution à l'instant précédent $t-1$, la distribution de filtrage s'écrit :

$$\begin{aligned} p(\boldsymbol{\alpha}_t | \mathbf{y}_{1:t}) &\propto p(\mathbf{y}_t | \mathbf{x}_t) p(\mathbf{x}_t, \boldsymbol{\lambda}_t | \boldsymbol{\mu}_t) \int p(\boldsymbol{\mu}_t | \boldsymbol{\mu}_{t-1}) q(\boldsymbol{\alpha}_{t-1}) d\boldsymbol{\alpha}_{t-1} \\ &\propto p(\mathbf{y}_t | \mathbf{x}_t) p(\mathbf{x}_t, \boldsymbol{\lambda}_t | \boldsymbol{\mu}_t) \int p(\boldsymbol{\mu}_t | \boldsymbol{\mu}_{t-1}) q(\boldsymbol{\mu}_{t-1}) d\boldsymbol{\mu}_{t-1} \end{aligned} \quad (13)$$

où seule l'intégration par rapport à $\boldsymbol{\mu}_{t-1}$ est utilisée grâce à la forme séparable de $q(\boldsymbol{\alpha}_{t-1})$. Ceci est aussi l'un des points clefs de cette invention où la dépendance temporelle est limitée à la fonctionnelle d'une seule composante. Dans un contexte décentralisé, la communication entre 2 unités en charge de la mise à jour de la distribution de filtrage se trouve limitée à l'envoi de $q(\boldsymbol{\mu}_{t-1})$ qui représente ainsi la statistique suffisante. De plus, un simple calcul permet de montrer que cette fonctionnelle est une gaussienne et donc que la communication entre deux noeuds leaders successifs se résume à l'envoi d'une moyenne et d'une covariance. Ainsi, l'approche classique [3] consistant à mettre à jour dans un premier temps les densités de probabilité et à les approximer dans un deuxième temps n'est plus nécessaire. Ce traitement conjoint des données et de l'approximation des statistiques suffisantes est le point clé de notre démarche dans ce projet.

0.4 Résultats de simulation

Dans cette section, nous illustrons l'efficacité et la robustesse de l'algorithme variationnel à noyau (DD-VF) pour le suivi d'une cible mobile dans un réseau de capteurs sans fil, en le comparant par rapport au filtre variationnel classique avec un modèle d'observation connu. Nous avons considéré une trajectoire composée par 2 sinusoides dans un champ 2D (voir figure 6), pour une durée de 200 unités de temps. Un changement brusque de trajectoire est simulé à l'instant $t_a = 100$ afin de tester la capacité de l'algorithme à poursuivre la cible dans un cas difficile de trajectoire discontinue. Un ensemble de 500 capteurs est déployé d'une manière aléatoire sur une surface de $120m \times 120m$. Chaque capteur possède une couverture de 20m. A chaque instant t , les matrices connues \mathbf{K}_{tt} et \mathbf{K}_{tp} (entrées de l'algorithme) sont simulées selon le modèle stationnaire suivant :

$$\begin{cases} K_{tt}(i, j) = \exp\{-\|\mathbf{s}_i^{(t)} - \mathbf{s}_j^{(t)}\|/2\sigma^2\} + \epsilon_{ij}^{(t)}, 1 \leq i, j \leq n \\ K_{tp}(j) = \exp\{-\|\mathbf{s}_j^{(t)} - \mathbf{x}_t^*\|/2\sigma^2\} + \epsilon_j^{(t)}, 1 \leq j \leq n \end{cases} \quad (14)$$

où $\mathbf{s}_m^{(t)} = (s_1^m, s_2^m)$ et $\mathbf{x}_t^* = (x_1, x_2)$ sont les positions du capteur activé et de la cible à l'instant t . Le paramètre σ est fixé à 10 et $\epsilon_{ij}^{(t)}$ désigne le bruit causé par l'erreur de modélisation, le bruit de mesure et les interférences des signaux voisins. La variance du bruit dépend de la distance inter-capteurs et capteur-cible.

La matrice \mathbf{G}_{tt} est formée par les produits scalaires $\langle \mathbf{s}_i^{(t)}, \mathbf{s}_j^{(t)} \rangle$, pour $i, j = 1, \dots, n$, entre les positions connues des capteurs.

Le nombre maximum des capteurs activés à chaque instant est fixé à 10. Le protocole de sélection est basé sur le calcul de la distribution prédictive de la position de la cible. On montre dans [2], en se basant sur un calcul variationnel, que cette distribution est gaussienne, ce qui rend le protocole très simple à implémenter. Plus de détails sur les protocoles de sélection des capteurs ainsi que la comparaison de leurs performances se trouvent dans [7] et [9].

Les hyperparamètres du modèle de transition (modèle a priori) ont été fixés comme suit :

$$\bar{\boldsymbol{\lambda}} = 10^{-2} \mathbf{I}, \quad \bar{n} = 1, \quad \bar{\mathbf{S}} = 10^2 \mathbf{I},$$

qui induisent un modèle *a priori* très flexible et non informatif. Il convient de noter que dans les méthodes classiques de tracking, un modèle de transition linéaire se basant sur la vitesse et l'accélération est souvent utilisé. Dans ce travail, le modèle est très général, ce qui le rend utilisable dans d'autres applications.

L'algorithme de tracking proposé est appliqué pour estimer en ligne la position de la cible sans aucune connaissance du modèle d'observation (14). La figure 6 montre la trajectoire estimée de la cible. La position de la cible est estimée par l'espérance a posteriori ($\hat{\mathbf{x}}_t = \langle \mathbf{x}_t \rangle$) selon la densité de filtrage approchée $q(\mathbf{x}_t)$. On note bien la précision du tracking évaluée selon l'erreur quadratique moyenne $mse = \sum_t \|\mathbf{x}_t - \hat{\mathbf{x}}_t\|^2 / T = 0.29$. Sur la même figure 6, les 10 capteurs sélectionnés pour traiter l'information sont marqués par des cercles pour 4 instants : $t = 40, t = 80, t = 160$ et $t = 190$. On note la capacité de l'algorithme pour sélectionner les capteurs les plus pertinents en se basant sur une forme compacte gaussienne de la distribution prédictive.

Pour des fins de comparaison, l'algorithme variationnel classique (VF) [2] est appliqué pour suivre la cible dans les mêmes conditions ci-dessus, avec 10 capteurs sélectionnés à chaque instant. L'algorithme VF est appliqué en utilisant le vrai modèle d'observation (14) ayant servi à générer les données de simulation. La figure 7 montre le résultat du tracking de la cible mobile. L'erreur quadratique moyenne est évaluée à $mse = 1.3$. On note bien que l'al-

gorithme variationnel classique est moins performant que l'algorithme proposé n'utilisant pas le modèle d'observation. Cet écart de performance peut s'expliquer par le fait que l'algorithme proposé exploite en plus les données issues des RSSI inter-capteurs. Cet écart de performance peut se creuser davantage lorsque l'algorithme classique utilise un modèle d'observation erroné ou lorsque le champs de déploiement des capteurs est non stationnaire. En effet, l'algorithme proposé est basé sur les données d'apprentissage récoltées par les capteurs pour estimer un modèle d'observation local, ce qui le rend très robuste par rapport à la non stationnarité de ce modèle ou l'influence de la dégradation des capteurs.

0.5 Conclusion

Dans ce chapitre, nous avons décrit, d'une manière synthétique, l'un de nos travaux sur le traitement décentralisé de l'information dans les réseaux de capteurs sans fil. Ce travail présente une nouvelle méthodologie exploitant à la fois la consistance de l'approche bayésienne et aussi une technique d'apprentissage afin de remédier à l'absence d'un modèle fiable d'observation. L'exploitation des signaux échangés entre les capteurs de positions connues est formulée par un problème de régression (complétion) matricielle dans un espace de Hilbert à noyau reproduisant. La forme gaussienne de la fonction de vraisemblance obtenue, permet une mise en oeuvre rapide et efficace du filtre variationnel. Cette technique peut s'appliquer aussi bien pour le tracking de cible que pour l'auto-localisation des capteurs mobiles.

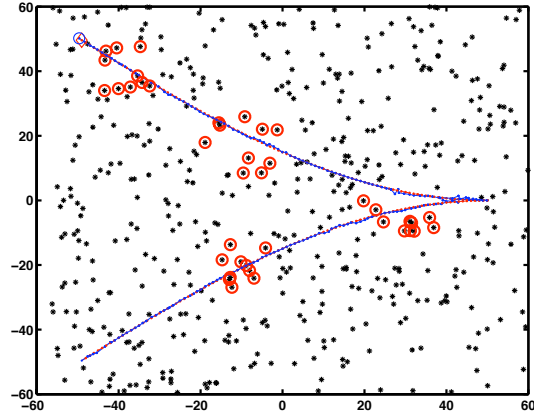


FIGURE 6 – Filtrage variationnel sans modèle dans un réseau de capteurs coopératifs : Les positions estimées sont en bleu et les vraies positions en rouge. Les 10 capteurs sélectionnés sont tracés en cercles, pour 4 instants choisis, $t = 40, t = 80, t = 160$ et $t = 190$.

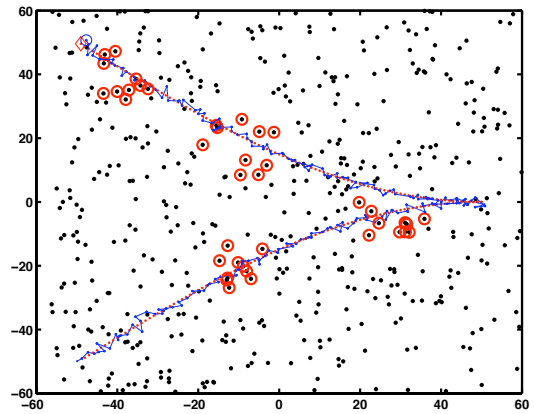


FIGURE 7 – Filtrage variationnel classique dans un réseau de capteurs coopératifs : Les positions estimées sont en bleu et les vraies positions en rouge. Les 10 capteurs sélectionnés sont tracés en cercles, pour 4 instants choisis, $t = 40, t = 80, t = 160$ et $t = 190$.

Bibliographie

- [1] “Special issue on self-organizing distributed collaborative sensor networks”, *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 4, 2005.
- [2] H. Snoussi and C. Richard, “Ensemble learning online filtering in wireless sensor networks”, in *IEEE ICCS International Conference on Communications Systems*, 2006.
- [3] A. Ihler, J. Fisher III, and A. Willsky, “Particle filtering under communications constraints”, in *Proc. Statistical Signal Processing (SSP) 2005*, 2005.
- [4] X. Sheng, Y. Hu, and P. Ramanathan, “Distributed particle filter with GMM approximation for multiple targets localization and tracking in wireless sensor network”, in *Information Processing in Sensor Networks, 2005. IPSN 2005*, April 2005, pp. 181–188.
- [5] N. Patwari and A. O. Hero, “Manifold learning algorithms for localization in wireless sensor networks”, in *IEEE ICASSP*, May 2004, vol. 3, pp. 857–860.
- [6] P. Djurić, M. Vemula, and M. Bugallo, “Tracking with particle filtering in tertiary wireless sensor networks”, in *ICASSP*, Philadelphia, PA, USA, March 18-23 2005.
- [7] J. Teng, H. Snoussi, and C. Richard, “Prediction-based proactive cluster target tracking protocol for binary sensor networks”, in *The 7th IEEE International Symposium on Signal Processing and Information Technology*, 2007.
- [8] J. Teng, H. Snoussi and C. Richard, “Decentralized Variational Filtering for Target Tracking in Binary Sensor Networks.” *IEEE Transactions on Mobile Computing*, 2010.
- [9] J. Teng, H. Snoussi and C. Richard, “Prediction-based Cluster Management for Target Tracking in Wireless Sensor Networks” *Journal of Wireless Communications and Mobile Computing*, 2010

- [10] J. Teng, H. Snoussi and C. Richard, “ Collaborative Multi-target Tracking in Wireless Sensor Networks. ” *International Journal of Systems Science*, 2011.
- [11] B. Schölkopf and A. Smola, *Learning with Kernels : Support Vector Machines, Regularization, Optimization, and Beyond*, MIT Press, Cambridge, MA, USA, 2001.
- [12] Y. Yamanishi and J.-P. Vert, “Kernel matrix regression”, Tech. Rep. <http://arxiv.org/abs/q-bio/0702054v1>, 2007.
- [13] C. Hue, J.-P. Le Cadre, and P. Pérez, “Sequential Monte Carlo methods for multiple target tracking and data fusion”, *IEEE Transactions on Signal Processing*, vol. 50, pp. 309–325, 2002.
- [14] O. Barndorff-Nielsen, “Exponentially decreasing distributions for the logarithm of particle size”, in *Proc. Roy. Soc.*, London, 1977, vol. 353, pp. 401–419.
- [15] J. Vermaak, N. Lawrence, and P. Perez, “Variational inference for visual tracking”, in *Conf. Computer Vision and Pattern Recog, CVPR’03*, June 2003.
- [16] A. Doucet, S. Godsill, and C. Andrieu, “On sequential Monte Carlo sampling methods for Bayesian filtering”, *Statistics and Computing*, vol. 10, no. 3, pp. 197–208, 2000.
- [17] A. Doucet, B. Vo, C. Andrieu, and M. Davy, “Particle filtering for multitarget tracking and sensor management”, in *Proc. Int. Conf. Inform. Fusion*, March 2002, vol. 1, pp. 474–481.