

Suivi de cibles à l'aide d'un réseau de capteurs

Adrien ICKOWICZ, Jean Pierre LE CADRE

IRISA / CNRS
Rennes, France

ickowicz@irisa.fr, lecadre@irisa.fr

Résumé – Le but de cet article est de présenter un nouvel algorithme pour le suivi de cibles à l'aide d'un réseau de capteurs binaires. L'article actuel se base sur des travaux développés dans [1]. Une nouvelle méthode de suivi est proposée, et ses performances pour le suivi d'un modèle de trajectoire classique sont présentées. La principale originalité vient de l'utilisation de méthodes de classification statistique (Séparateurs à Vastes Marge) pour la localisation de la cible.

Abstract – The aim of this paper is to present a new algorithm for target tracking within a binary sensor network. The present work is based on our previous results developed in [1]. A novel tracking method is proposed and its performance through a very classical trajectory model is evaluated. For a given target, this algorithm provides an estimation of its velocity and then of its position. The greatest improvements are made through a position correction and velocity analysis.

1 Introduction

La problématique principale des réseaux de capteurs consiste à utiliser la topologie de ces réseaux afin d'extraire et de traiter l'information la plus fiable possible. Ces capteurs pouvant être limités en terme de portée, précision, autonomie... L'intérêt d'utiliser une information binaire est qu'elle ne demande pas ou peu de traitement local, et coûte moins en énergie lors de la transmission. De même, la modélisation d'une erreur d'information est très facile. Il existe plusieurs type d'informations de ce genre, et nous avons décidé de nous concentrer sur l'information de mouvement : la cible s'approche t'elle ou s'éloigne t'elle du capteur [2] ? Nous présentons dans cet article une méthode d'estimation de la position et de la vitesse basée sur l'information spatiale et temporelle qui permet un suivi particulièrement efficace du déplacement d'une cible dans un réseau de capteurs binaires

2 Suivi de cible dans un réseau de capteurs binaires

Nous supposons que notre cible évolue suivant un modèle markovien :

$$\mathbf{x}_t | \mathbf{x}_{t-1} \sim \mathcal{N}(F_t \mathbf{x}_{t-1}, \mathbf{Q}_t) \quad (1)$$

pour tout $t = 1, 2, \dots$ avec $\mathcal{N}(\mu, \sigma^2)$ une loi normale de moyenne μ et de variance σ^2 . La position de départ de la cible est supposée inconnue. A chaque instant, les capteurs nous donnent une information binaire, signifiant si la cible s'approche ou s'éloigne dudit capteur. Nous connaissons dès lors la zone où se situe notre cible à chaque instant.

2.1 Problème de localisation de la cible

Notons A l'ensemble des capteurs dont la cible s'approche et B celui dont elle s'éloigne, et $C(A)$ et $C(B)$ leurs enveloppes convexes :

Proposition 1 $C(A) \cap C(B) = \emptyset$ et $\mathbf{x}_t \notin C(A) \cup C(B)$.

Donc notre cible se trouve entre ces deux espaces. De plus $C(A)$ et $C(B)$ étant deux espaces convexes disjoints, il existe un hyperplan séparateur. Dès lors, avec t_k la position du capteur k , nous pouvons écrire $\mathbf{t}_k = \lambda \mathbf{v}_t + \mu \mathbf{v}_t^\perp$, et nous avons :

$$\langle \mathbf{t}_k, \mathbf{v}_t \rangle = \lambda \|\mathbf{v}_t\|^2 > 0 \iff \lambda > 0. \quad (2)$$

Ce qui signifie que la droite générée par le vecteur \mathbf{v}_t^\perp , et passant par la position de la cible, sépare $C(A)$ et $C(B)$. Puis, considérant la base $\{\mathbf{v}_t, \mathbf{v}_t^\perp\}$, nous avons :

$$\begin{cases} \mathbf{t}_k \in A \iff \lambda \|\mathbf{v}_t\|^2 > \langle \mathbf{x}_t, \mathbf{v}_t \rangle, \\ \mathbf{t}_k \in B \iff \lambda \|\mathbf{v}_t\|^2 < \langle \mathbf{x}_t, \mathbf{v}_t \rangle. \end{cases} \quad (3)$$

Ainsi, dans la base $(\mathbf{v}_t, \mathbf{v}_t^\perp)$, la droite passant par le point $\left(\frac{\langle \mathbf{x}_t, \mathbf{v}_t \rangle}{\|\mathbf{v}_t\|^2}, 0 \right)$

et de direction \mathbf{v}_t^\perp sépare $C(A)$ et $C(B)$. Cela signifie que si nous estimons cette droite séparatrice (par une méthode que nous décrivons plus loin), nous obtenons une estimation de la direction du vecteur vitesse. Nous normalisons cet estimateur car nous n'avons pas encore d'idée de la valeur de la vitesse. Cette estimation intervient dans le prochain paragraphe.

2.2 Estimation de la vitesse directionnelle

Le but clairement défini par le précédent paragraphe étant d'estimer cette droite séparatrice, nous utilisons la méthode des Séparateurs à Vaste Marge (SVM) [3], largement utilisée dans des contextes de classification. Soient $\{(y_1, \mathbf{x}_1), \dots, (y_l, \mathbf{x}_l)\}$ ($y_i \in \{-1, 1\}$ et \mathbf{x}_i positions des capteurs). Le problème est dit linéairement séparable s'il existe un vecteur \mathbf{w} et un scalaire b tels que :

$$\begin{cases} \langle \mathbf{w}, \mathbf{x}_i \rangle + b \geq 1 & \text{si } y_i = 1, \\ \langle \mathbf{w}, \mathbf{x}_i \rangle + b \leq -1 & \text{si } y_i = -1. \end{cases} \quad (4)$$

Soit $\mathcal{H}(\mathbf{w}, b) \triangleq \{\mathbf{x} | \langle \mathbf{w}, \mathbf{x} \rangle + b = 0\}$ (\mathbf{w} : vecteur normal) l'hyperplan séparateur optimal, nous définissons la marge comme la distance du capteur le plus proche à l'hyperplan, ce qui nous donne $\text{marge} = \frac{1}{\|\mathbf{w}\|}$. Dès lors, maximiser la marge revient au problème suivant :

$$\begin{cases} \min_{\mathbf{w}, b} \tau(\mathbf{w}) \stackrel{\delta}{=} \|\mathbf{w}\|^2, \\ \text{s.t. } : y_i (\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 \quad \forall i = 1, \dots, l \quad y_i = \pm 1. \end{cases} \quad (5)$$

Le problème dual peut être résolu par des méthodes de programmation quadratiques classiques. Le cas un peu moins parfait revient à considérer qu'il ne peut y avoir de séparation sans erreur, et nous remplaçons les contraintes de l'équation 5 par :

$$y_i (\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \dots, l. \quad (6)$$

Ceci nous permet donc d'avoir une estimation de notre vecteur vitesse directionnel instant après instant.

2.3 Intégration Multi-Période

Il nous reste désormais à estimer la valeur de la vitesse de notre cible ainsi que sa position, ce qui n'est pas un moindre problème. Nous avons décidé de prendre en compte les informations passées pour construire des estimateurs de position et de vitesse. Pour cela, nous avons développé un algorithme en deux étapes. Lors de la première, nous proposons une correction le long de la direction de vitesse estimée, appelée λ_t (qui s'avère estimer la norme de la vitesse), puis lors de la deuxième étape, une correction orthogonale est ajoutée, appelée θ_t (qui estime la position). Voir la figure 1 pour une compréhension graphique de l'algorithme.

2.3.1 Mise en équation et algorithme

Pour calculer la correction longitudinale, nous considérons tout simplement que la cible doit se trouver entre les deux zones convexes de capteurs indiquant l'approchement ou l'éloignement de ladite cible.

Nous classons alors $vs_i^{(-)} = \langle \hat{\mathbf{v}}_t, t_i^{(-)} \rangle$ (respectivement $vs_i^{(+)} = \langle \hat{\mathbf{v}}_t, t_i^{(+)} \rangle$) les projections orthogonales des positions des capteurs indiquant (+) et des capteurs indiquant (-). Dès lors,

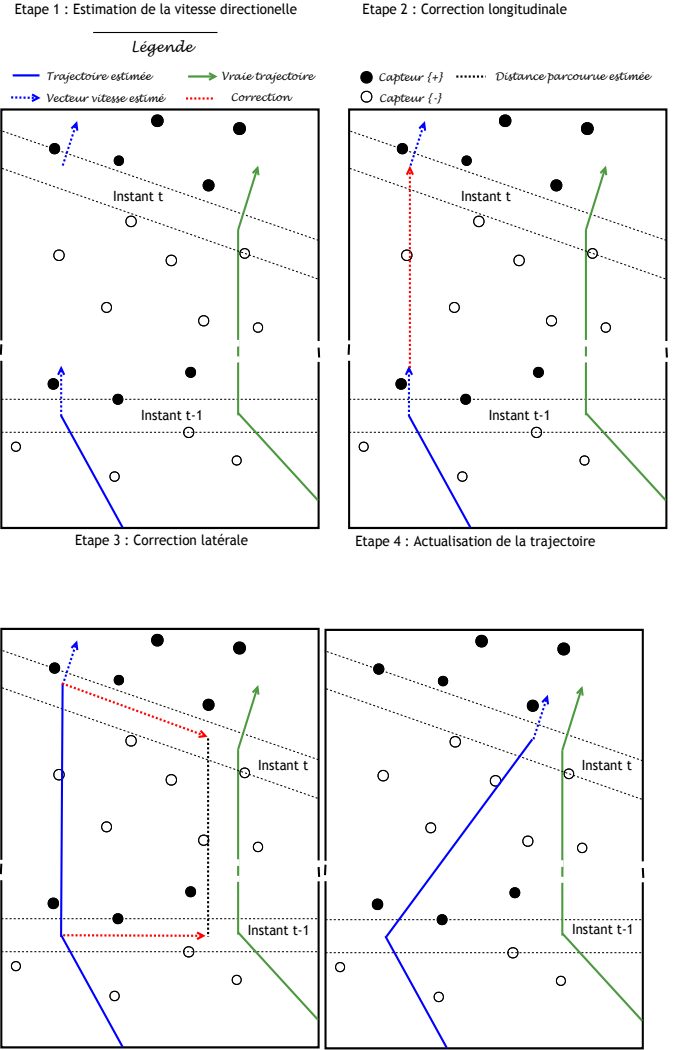


FIG. 1 – Les étapes de l'algorithme.

suivant un raisonnement géométrique très simple, $\langle \hat{\mathbf{v}}_t, \hat{\mathbf{X}}_t \rangle$ devrait être entre $vs_{max}^{(-)}$ et $vs_{min}^{(+)}$. D'où les expressions mathématiques suivantes :

$$\lambda_t = \frac{vs_{moy}^{(+,-)} - \langle \hat{\mathbf{v}}_t, \hat{\mathbf{x}}_{t-1} \rangle}{\langle \hat{\mathbf{v}}_t, \hat{\mathbf{v}}_{t-1} \rangle}, \quad vs_{moy}^{(+,-)} = \frac{vs_{max}^{(-)} + vs_{min}^{(+)}}{2}. \quad (7)$$

Concernant la correction latérale, l'hypothèse que nous faisons peut être plus discutable, mais la qualité des résultats obtenus défend ce choix. Nous supposons donc que d'un instant à l'autre, la variation de vitesse est limitée. Il s'agit de la valeur de la vitesse, et non de son orientation. Et même si nous pourrions également limiter le changement de direction, il ne nous est pas apparu utile de l'inclure dans les contraintes. Cette hypothèse (que l'on retrouve souvent en analyse classique avec les fonction lipschitzienne, ou encore en statistique des processus avec les processus à variation bornée) se retrouve mathématiquement dans cette expression :

$$\langle \hat{\mathbf{x}}_t^{corr} + \theta_t \hat{\mathbf{v}}_t^\perp - (\hat{\mathbf{x}}_{t-1} + \theta_t \hat{\mathbf{v}}_{t-1}^\perp); \hat{\mathbf{v}}_{t-1} = m_{t,k} \quad (8)$$

Ici $m_{t,k}$ correspond à la moyenne des k dernières valeurs de vi-

Algorithm 1 Algorithme de suivi pour un réseau de capteurs binaires

Require: $\forall t, S_t$

- 1: $\hat{X}_0 \sim \mathcal{U}(S_0)$
 - 2: $\hat{V} \leftarrow \mathcal{N}(m, \sigma^2)$
 - 3: **for** $t = 1$ à T **do**
 - 4: $\hat{v}_t \leftarrow SVM(S_t)$
 - 5: $\lambda_t \leftarrow \frac{vs(\hat{V}) - \langle \hat{v}_t, \hat{x}_{t-1} \rangle}{\langle \hat{v}_t, \hat{v}_{t-1} \rangle}$
 - 6: $\theta_t \leftarrow \frac{|\hat{V}| - \lambda_t}{\langle \hat{v}_t^\perp; \hat{v}_{t-1} \rangle}$
 - 7: $\hat{X}_t \leftarrow \lambda_t(\hat{V})\hat{v}_t + \theta_t(\hat{V})\hat{v}_t^\perp$
 - 8: $\hat{V} \leftarrow \hat{X}_t - \hat{X}_{t-1}$
 - 9: **end for**
 - 10: **return** \hat{X}, \hat{V}
-

tesse. Cela nous donne finalement une expression formelle de la correction latérale :

$$\theta_t = \frac{m_{t,k} - \lambda_t}{\langle \hat{v}_t^\perp; \hat{v}_{t-1} \rangle} \quad (9)$$

L'estimation finale de la position devient alors :

$$\hat{x}_t^{fin} = \lambda_t \hat{v}_t + \theta_t \hat{v}_t^\perp. \quad (10)$$

Enfin, une dernière étape vient clore le processus d'estimation. Clairement, d'après les expressions précédentes, la correction latérale dépend de la qualité des estimations précédentes de la vitesse de déplacement de la cible. Classiquement, plus cette estimation est précise, meilleure est la correction par la suite. Ainsi, notre estimation de la position se retrouve conjointe à une bonne estimation de la vitesse, ce qui n'est pas une grande surprise quand on parle de suivi de cible. Considérée instant après instant, l'estimation de la position s'améliore, et il nous est donc apparu naturel que si nous pouvions retranscrire ces corrections aux estimations passées, la vitesse estimée s'en retrouverait améliorée, et notre algorithme plus efficace. Ainsi, les positions sont actualisées comme suit :

$$\forall j < t : \hat{z}_j = \hat{x}_j^{fin} + \sum_{i=j+1}^t \theta_i \hat{v}_i^\perp. \quad (11)$$

2.3.2 Résultats

Nous présentons les résultats de simulations pour une cible se déplaçant dans un espace de 200mx200m, couvert par 81 capteurs. La variance de la trajectoire est faible, et la durée du suivi est de 30 secondes. Un résultat de suivi est présenté sur la figure 2. En rouge la vraie trajectoire, et en vert l'estimation. La figure en dessous présente l'erreur quadratique moyenne de l'estimateur de la position pour trois topologies de réseau différentes.

L'initialisation de la position de la cible présente une étape supplémentaire de l'algorithme. Soit nous connaissons la position de départ de la cible, auquel cas le problème d'initialisation n'existe plus. Soit nous ne la connaissons pas, et dans ce cas

la, La méthode est assez simple. A l'instant initial, nous possédons deux zones de capteurs, définies par l'approchement ou l'éloignement. Nous prenons dès lors une position dans l'espace entre les deux zones de façon complètement aléatoire, et sans loi a priori. Dans le cas présent, étant donné le nombre de capteurs disponibles, l'erreur d'initialisation aurait été assez petite.

Concernant l'erreur quadratique moyenne, celle-ci décroît ra-

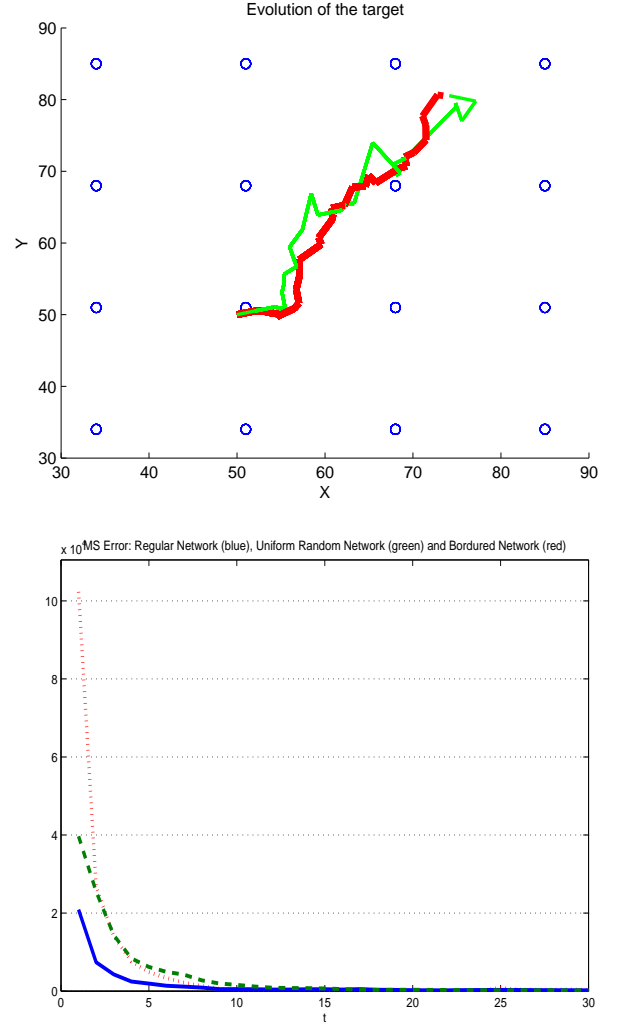


FIG. 2 – Suivi de cible à l'aide de capteurs binaires.

pidement en fonction du temps. Nous avons prolongé la durée du suivi, (figure non présentée) et contrairement à ce que pourrait laisser penser la figure présentée, l'EQM ne converge pas vers 0. En effet, il existe une quantité non dénombrable de scénario (au moins tous ceux dont la trajectoire longe un axe du réseau, pour le réseau régulier) pour lesquels passé la distance entre deux capteurs, nous ne pouvons améliorer l'estimation de la position de la cible.

Nous avons effectué une comparaison entre plusieurs topographies de réseaux, et nous les avons soumis à comparaison. Et

la conclusion semble évidente. Aucun d'entre eux ne semble strictement préférable, au sens de l'EQM de la position tout du moins.

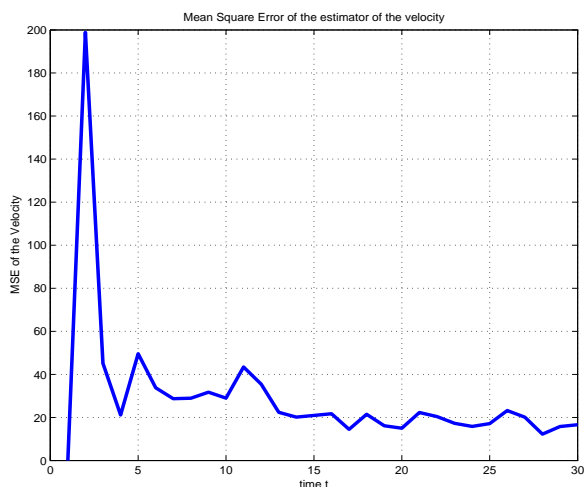


FIG. 3 – Erreur Quadratique Moyenne sur l'estimation de la vitesse.

3 Conclusion

Nous avons développé une nouvelle méthode pour le suivi de cible dans un réseau de capteurs binaires spécifiques. Bien que l'information instantanée soit relativement faible, nous avons réussi à utiliser l'information passée pour pouvoir estimer le plus finement possible à la fois la position et la vitesse de la cible. Ceci est d'autant plus remarquable que les hypothèses faites sur le mouvement de la cibles ne sont absolument pas restrictives. De plus, notre algorithme reste rapide et fiable. Il serait intéressant d'analyser l'effet de la fréquence d'acquisition des données sur nos résultats de suivi.

Toutefois, des points importants restent sans réponse, et demande d'avantage de travaux. En particulier, nous n'avons guère accordé d'importance à la modélisation de la vitesse. Nous nous sommes concentrés sur les différents facteurs correctifs, mais nous n'avons pas pris en compte la modélisation réelle de la vitesse, à l'aide d'une chaîne de markov cachée, par exemple. Ceci pourrait certainement aider à améliorer la qualité de l'estimation. Ce point revêt toute son importance lorsque l'on considère l'extension de l'algorithme pour du suivi multicible. Si chaque cible a son modèle de vitesse, lorsque les positions se retrouveront proches, il nous sera certainement plus facile de ne pas inverser les trajectoires, et d'optimiser l'association des mesures.

De plus, notre algorithme reste complètement déterministe, et ce même si le mouvement de la cible est stochastique. Il pourrait par exemple être de bon ton d'utiliser un filtrage particulière sur l'estimation de la position à l'aide des correcteurs, ceci afin d'affiner notre estimation et d'obtenir (sans certitudes) une

vitesse de convergence de l'EQM plus importante.

Ce sont autant de pistes en cours de développement. Même si nos travaux sont préliminaires, nous croyons fermement que notre algorithme de suivi permettra du suivi multicible et en simplifiera les problèmes d'association inhérents.

Références

- [1] A. ICKOWICZ, J.-P. LE CADRE, A new method for target trajectory estimation within a binary sensor network. *Proc. of the 10th European Conference on Computer Vision : Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications Workshop*, Oct 2008.
- [2] J. ASLAM, Z. BUTLER, F. CONSTANTIN, V. CRESPI, G. CYBENKO, D. RUS : Tracking a moving object with a binary sensor network. *Proc. of the 1st international Conference on Embedded Networked Sensor Systems*, Nov 2005, pp. 150–161.
- [3] C. CORTES, V. VAPNIK : Support-Vector Networks. *Machine Learning*, **20**, 1995, pp. 273–297.
- [4] X. WANG and B. MORAN : Multitarget Tracking Using Virtual Measurements of Binary Sensor Networks. *Proc. of the 9-th Int. Conf. on Information Fusion*, Jul. 2006.