Radar MIMO massif cognitif : détection et suivi conjoints de cibles dans des perturbations inconnues

Imad BOUHOU[1,3] Stefano FORTUNATI[2] Leila GHARSALLI[3] Alexandre RENAUX[1]

[1]Université Paris-Saclay, CNRS, CentraleSupelec, Laboratoire des Signaux et Systèmes, 91190, Gif-sur-Yvette, France

[2]SAMOVAR, Télécom SudParis, Institut Polytech- nique de Paris, 91120, Palaiseau, France

[3]DR2I-IPSA, 94200, Ivry sur Seine, France

Résumé – Ce travail présente un radar cognitif utilisant un processus de décision markovien partiellement observable (POMDP) pour détecter et suivre des cibles mobiles dans un environnement bruité inconnu. Basé sur la technologie radar MMIMO (massif multiple-input multiple-output), le système agit comme un agent intelligent optimisant ses actions pour maximiser la probabilité de détection (P_D) et améliorer l'estimation de l'état cible tout en maintenant un taux constant de fausses alarmes (P_{FA}). Contrairement aux méthodes classiques, nous ne supposons que peu d'hypothèses concernant les perturbations et la cible est non-stationnaire. Nos simulations montrent que l'algorithme en ligne proposé surpasse la méthode SARSA (State-Action-Reward-State-Action) pour les systèmes radar MMIMO.

Abstract – This work presents a cognitive radar using a Partially Observable Markov Decision Process (POMDP) to detect and track moving targets in an unknown noisy environment. Based on MMIMO (massive multiple-input multiple-output) radar technology, the system acts as an intelligent agent optimizing its actions to maximize detection probability (P_D) and improve target state estimation while maintaining a constant false alarm rate (P_{FA}). Unlike conventional methods, we assume few perturbation assumptions, and the target is non-stationary. Our simulations show that the proposed online algorithm outperforms the SARSA (State-Action-Reward-State-Action) method for MMIMO radar systems.

1 Introduction

Le radar cognitif, introduit par Haykin dans [4], constitue une avancée majeure par rapport aux radars traditionnels. Inspiré du cycle perception-action biologique, il adapte en temps réel ses formes d'ondes en apprenant de son environnement, améliorant ainsi ses performances. Malgré son introduction il y a près de vingt ans, de nombreuses recherches portent encore sur ses applications pratiques. Les approches classiques, quant à elles, modélisent les perturbations comme des bruits Gaussiens additifs et appliquent le filtrage bayésien dans un espace distance-azimut-élévation pour estimer et prédire l'état de l'environnement.

Dans cette communication, nous proposons un cadre de radar cognitif reposant sur un système MMIMO. Celui-ci combine la robustesse et la propriété *Constant false alarm rate* (CFAR) du détecteur robuste de type de Wald [3] avec la capacité d'un algorithme basé sur l'apprentissage par renforcement (RL) afin de maximiser la probabilité de détection P_D tout en améliorant les performances de suivi. Pour prendre en compte la non-stationnarité du scénario, le problème est modélisé comme un processus de décision Markovien partiellement observable (POMDP) [5]. Le POMDP gère la prise de décision séquentielle dans des configurations dont les états sont cachés. Certains travaux antérieurs ont déjà exploité des solveurs POMDP en ligne pour traiter le suivi des cibles dans les applications radar [7]. Ce travail utilise l'algorithme *Partially Observable Monte-Carlo Planning* (POMCP) [8] et la robustesse du détecteur du type de Wald [3] pour développer un algorithme de suivi quasi optimal en présence d'une perturbation à statistiques inconnues.

2 Formulation du problème

Le modèle de signal du radar MMIMO considéré ici est le même que celui utilisé dans [3]. Nous considérons un radar MIMO colocalisé doté de N_T antennes émettrices et N_R antennes réceptrices, formant $N = N_T N_R$ canaux virtuels. Le champ de vision (FoV) est divisé en L angles $\{\theta_l; l = 1, \ldots, L\}$ et le système effectue T_{max} balayages. Pour un angle l à l'instant t, le problème de détection s'exprime par

$$\begin{aligned} H_0 &: \mathbf{y}_{t+1,l} = \mathbf{c}_{t+1,l}, \\ H_1 &: \mathbf{y}_{t+1,l} = \alpha_{t+1,l}\mathbf{v}_{t,l} + \mathbf{c}_{t+1,l}, \end{aligned} \tag{1}$$

où $\alpha_{t+1,l} \in \mathbb{C}$ est le coefficient de la Surface Équivalente Radar (SER). Le vecteur de perturbation $\mathbf{c}_{t+1,l} \in \mathbb{C}^N$ suit une distribution inconnue p_C. Nous supposons que les N composantes du vecteur de perturbation sont échantillonnées d'un processus aléatoire complexe circulaire $\{c_{t+1,l,n}, \forall n\}$. Nous supposons uniquement que sa fonction d'autocorrélation existe et décroît *au moins* à un taux polynomial [3]. Le vecteur $\mathbf{v}_{t,l} = (\mathbf{W}_t^T \mathbf{a}_T(\theta_l)) \otimes \mathbf{a}_R(\theta_l)$ est défini via la matrice d'onde $\mathbf{W}_t$ et les vecteurs directeurs $\mathbf{a}_T(\theta_l)$ et $\mathbf{a}_R(\theta_l)$ comme dans [3]. La matrice $\mathbf{W}_t$ est obtenue en maximisant l'énergie focalisée sur θ_l sous une contrainte de puissance totale fixe. Le test (1) repose sur la statistique robuste

$$\Lambda_{t+1,l} = 2|\hat{\alpha}_{t+1,l}|^2 \frac{||\mathbf{v}_{t,l}||^4}{\mathbf{v}_{t,l}^H \widehat{\mathbf{\Sigma}}_{t+1,l} \mathbf{v}_{t,l}}, \tag{2}$$

avec $\widehat{\mathbf{\Sigma}}_{t+1,l}$ un estimateur de la matrice de covariance de p_C et $\hat{\alpha}_{t+1,l} = (\mathbf{v}_{t,l}^H \mathbf{y}_{t+1,l})/||\mathbf{v}_{t,l}||^2$ est un estimateur du paramètre

$\alpha_{t+1,l}$. La statistique $\Lambda_{t+1,l}$ est comparée à un seuil λ déterminé afin de garantir un taux de fausses alarmes P_{FA} prédéfini, ce dernier est défini, selon [3], par $\lambda \underset{N\to+\infty}{\simeq} -2\ln(P_{FA})$. De manière analogue, la probabilité de détection P_D est estimée à l'aide de la fonction Q_1 de Marcum selon la formule suivante :

$$P_D \underset{N\to+\infty}{\simeq} Q_1\left(\sqrt{2|\hat{\alpha}_{t+1,l}|^2 \frac{||\mathbf{v}_{t,l}||^4}{\mathbf{v}_{t,l}^H \widehat{\mathbf{\Sigma}}_{t+1,l} \mathbf{v}_{t,l}}}, \sqrt{\lambda}\right). \quad (3)$$

3 Processus de décision markovien partiellement observable (POMDP)

Un POMDP [5] se définit par le tuple $(\mathcal{S}, \mathcal{O}, \mathcal{A}, \mathcal{P}, \Omega, \mathcal{R})$, où $\mathcal{S}$ est l'ensemble des états, $\mathcal{O}$ l'ensemble des observations, et $\mathcal{A}$ l'ensemble des actions disponibles. Les probabilités de transition $\mathcal{P}^a_{s,s'} = p(S_{t+1} = s' \mid S_t = s, A_t = a)$ définissent la probabilité de passer de l'état s à s' après avoir pris l'action a. De même, les probabilités d'observation $\Omega^a_{s',o} = p(O_{t+1} = o \mid S_{t+1} = s', A_t = a)$ décrivent la probabilité d'observer o après avoir exécuté a et atteint s'. Le gain associé à la transition est noté $\mathcal{R}^a_{s,s'}$.
Une histoire h_t est définie par $h_t = (a_0, o_1, \ldots, a_{t-1}, o_t)$, représentant la séquence des actions effectuées et des observations obtenues jusqu'à l'instant t. L'objectif de l'agent est d'apprendre une politique $\pi(a \mid h) = p(A_t = a \mid H_t = h)$ qui associe à chaque histoire une distribution de probabilité sur les actions. Par ailleurs, l'agent peut construire un état de croyance $b(s \mid h) = p(S_t = s \mid H_t = h)$. La fonction de valeur $V_\pi(h)$ représente le gain espéré en suivant la politique π à partir de l'histoire h, actualisé par un facteur $\gamma \in (0, 1)$:

$$V_\pi(h) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \mathcal{R}^{a_t}_{s_t,s_{t+1}} \,\middle|\, \begin{array}{l} h_0 = h; s_0 \sim b(.|h_0) \\ t \geq 0 : a_t \sim \pi(.|h_t) \\ t \geq 0 : s_{t+1} \sim \mathcal{P}^{a_t}_{s_t} \\ t \geq 0 : o_{t+1} \sim \Omega^{a_t}_{s_{t+1}} \\ h_{t+1} = (h_t, a_t, o_{t+1}) \end{array}\right], \quad (4)$$

La fonction de valeur optimale est alors définie par $V^*(h) = \max_\pi V_\pi(h)$. Enfin, lors de l'exécution d'une action a suivie de l'observation o, l'état de croyance se met à jour selon

$$b(s' \mid h, a, o) \propto \Omega^a_{s',o} \sum_s \mathcal{P}^a_{s,s'} b(s \mid h). \quad (5)$$

4 L'algorithme POMCP

L'algorithme POMCP [8] est un algorithme de planification en ligne basé sur les arbres de recherche Monte-Carlo conçu pour les grands POMDPs lorsque les actions et les observations sont discrètes. Il s'agit d'une extension de l'algorithme *Upper Confidence bounds applied to Trees* (UCT) [6] aux POMDPs, en construisant un Processus de décision markovien (MDP) dont les états sont définis par des historiques.
Le POMCP utilise un générateur en boîte noire $\mathcal{G}(s, a) = (s', o, r)$, où $r = \mathcal{R}^a_{s,s'}$, au lieu de connaître explicitement les distributions $\mathcal{P}$ et Ω. L'agent exécute N_{sim} simulations via une recherche arborescente et calcule la valeur de l'historique actuel $V(h)$ à la racine de l'arbre. Le processus commence par échantillonner un état à partir de l'ensemble de croyance B (approximation de la croyance $b(.|h)$) à la racine de l'arbre, puis sélectionne les actions maximisant le critère *Upper Confidence Bound* (UCB1), défini par : $Q^{UCT}(h, a) = Q(h, a) + c\sqrt{\log(N(h))/N(h, a)}$ où c est un hyperparamètre réglé pour équilibrer l'exploration et l'exploitation et $Q(h, a)$ est la fonction de valeur d'action. Lorsqu'une feuille est atteinte, un nouveau nœud est ajouté, à partir duquel commence l'étape de *rollout*, consistant à exécuter une simulation depuis le nœud nouvellement ajouté en suivant une politique de *rollout* π_{rollout}.

Comme indiqué dans le Théorème 1 de [8], lorsque le nombre de simulations augmente, la fonction de valeur $V(h)$ calculée par le POMCP à la racine de l'arbre converge en probabilité vers la fonction de valeur optimale $V^*(h)$. À la fin des simulations, l'agent choisit l'action $a^* = \arg\max_{a\in\mathcal{A}} Q(h, a)$. Après avoir exécuté l'action optimale et observé o, la mise à jour de l'ensemble de croyance s'effectue de la manière suivante : l'algorithme sélectionne une particule aléatoire s dans B, génère (s', o', r) en utilisant $\mathcal{G}(s, a)$, et compare o' à o. Si $o' = o$, alors s' est ajouté au nouvel ensemble de croyances B'. Ce processus est répété jusqu'à ce que le nouvel ensemble B' contienne N_p particules.

5 Le radar cognitif en tant que POMDP

Dans cette section, nous montrons comment intégrer les définitions de POMDP dans le cadre radar et présentons l'algorithme complet.

5.1 L'espace des actions

L'action correspond à l'ensemble des matrices d'onde possibles parmi lesquelles le radar peut choisir. La résolution angulaire du radar étant divisée en L intervalles d'angle, il dispose de L actions, c'est-à-dire qu'il peut sélectionner une matrice d'onde parmi L afin de concentrer toute l'énergie sur un intervalle précis. À l'instant t, le radar choisit un intervalle $l \in \{1, 2, \ldots, L\}$ associé à l'angle θ_l, puis la matrice d'onde $\mathbf{W}_t$ est la racine carrée de la matrice $\frac{P_T}{N_T}\mathbf{a}^*_T(\theta_l)\mathbf{a}^T_T(\theta_l)$.

5.2 L'espace d'états

L'espace d'états comprend les positions et les vitesses possibles de la cible. À l'instant t, l'état est défini par $\mathbf{s}_t = [x_t, V_{x,t}, y_t, V_{y,t}]^T$, où $[x_t, y_t]$ et $[V_{x,t}, V_{y,t}]$ représentent respectivement les vecteurs de position et de vitesse de la cible. Le modèle cinématique pour la cible est donné par l'équation d'état : $\mathbf{s}_{t+1} = \mathbf{A}\mathbf{s}_t + \mathbf{G}\mathbf{w}_t$, où $\mathbf{A}$ est la matrice de transition d'état en bloc : $\mathbf{A} = \begin{bmatrix} \mathbf{A}_b & \mathbf{0}_{2\times2} \\ \mathbf{0}_{2\times2} & \mathbf{A}_b \end{bmatrix}$, tel que $\mathbf{A}_b = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}$. Le terme $\mathbf{G}\mathbf{w}_t$ représente le bruit, et la matrice $\mathbf{G}$ peut être écrite en forme de blocs comme suit : $\mathbf{G} = \begin{bmatrix} \mathbf{G}_b & \mathbf{0}_{2\times1} \\ \mathbf{0}_{2\times1} & \mathbf{G}_b \end{bmatrix}$, tel que $\mathbf{G}_b = \begin{bmatrix} \Delta t^2/2 \\ \Delta t \end{bmatrix}$. Le bruit est modélisé par $\mathbf{w}_t \sim \mathcal{N}\left(\mathbf{0}_2, \sigma_s^2 \mathbf{I}_2\right)$, où σ_s désigne l'écart-type du bruit.

5.3 L'espace d'observations

À l'instant t, le radar choisit une action a associée à un intervalle angulaire l, ce qui permet de calculer le vecteur $\mathbf{v}_{t,l}$. Ensuite, lors d'une détection, le radar récupère l'observation suivante :

$$o_{t+1} = \begin{cases} |\hat{\alpha}_{t+1,l}| & \text{si } \Lambda_{t+1,l} \geq \lambda, \\ \emptyset & \text{sinon.} \end{cases} \quad (6)$$

Classiquement, le paramètre (SER) $|\alpha_{t+1,l}|$ est inversement proportionnel à R_{t+1}^2, où R_{t+1} représente la distance entre la cible et le radar. De plus, comme démontré dans [3], l'estimation $\hat{\alpha}_{t+1,l}$ suit asymptotiquement une loi normale complexe

$$(\hat{\alpha}_{t+1,l} - \alpha_{t+1,l})/\widehat{\sigma}_{t,l} \underset{N\to\infty}{\sim} \mathcal{CN}(0,1), \quad (7)$$

avec $\widehat{\sigma}_{t,l} = (\sqrt{\mathbf{v}_{t,l}^H \widehat{\mathbf{\Sigma}}_{t+1,l} \mathbf{v}_{t,l}})/||\mathbf{v}_{t,l}||^2$.
Le POMCP est conçu pour des observations discrètes, il faut donc discrétiser l'espace d'observation (actuellement continu) en choisissant un pas de discrétisation β_l. En se basant sur l'approximation de la distribution de $|\hat{\alpha}_{t+1,l} - \alpha_{t+1,l}|^2$ par une loi exponentielle $\text{Exp}(\widehat{\sigma}_{t,l}^2)$ pour $N \to \infty$, le pas β_l est défini par la condition $\Pr\{\big|\, |\hat{\alpha}_{t+1,l}| - |\alpha_{t+1,l}| \,\big| < \beta_l\} \geq 0.95$. Il suffit que β_l vérifie la condition $\Pr\{\big|\, \hat{\alpha}_{t+1,l} - \alpha_{t+1,l} \,\big|^2 < \beta_l^2\} = 0.95$, ce qui conduit à $\beta_l = \sqrt{3}\,\widehat{\sigma}_{t,l}$.

5.4 La fonction de récompense

La fonction de récompense doit encourager le radar à détecter et suivre la cible dans l'environnement. Dans la définition du POMDP, la fonction de récompense dépend de l'état courant $\mathbf{s}$, de l'action entreprise a, et de l'état suivant $\mathbf{s}'$. L'action a consiste à choisir un intervalle d'angle θ_a où la cible se trouvera à l'avenir.
Notons par $\theta_{\mathbf{s}'}$ l'intervalle d'angle réel futur de la cible. Pour encourager une prédiction précise de la position de la cible, la fonction de récompense est choisie comme suit :

$$\mathcal{R}_{s,s'}^a = \mathbf{1}\{\theta_a = \theta_{\mathbf{s}'}\}. \quad (8)$$

On netera que la fonction de récompense ici ne dépend pas de l'état courant $\mathbf{s}$.

5.5 Modèle de simulation

L'algorithme POMCP [8] requiert un générateur en boîte noire $\mathcal{G}(s,a) = (s',o,r)$ pour exécuter des simulations dans la recherche arborescente. Dans ce travail, la perturbation de bruit p_C étant inconnue, les probabilités d'observation le sont également, rendant l'utilisation directe du POMCP impossible. Toutefois, en s'appuyant sur la distribution asymptotique de l'estimation $\hat{\alpha}_{t+1,l}$, un générateur $\mathcal{G}(\mathbf{s}_t, a_t)$ approprié peut être mis en place pour permettre l'utilisation du POMCP et du filtre à particules afin d'assurer la détection et le suivi d'une cible dans l'environnement.

6 Évaluation expérimentale

Le modèle de perturbation utilisé ici est basé sur un processus auto-régressif (AR) d'ordre p.

Algorithme 1 : Générateur $\mathcal{G}(\mathbf{s}_t, a_t)$

Input : $\mathbf{s}_t = (x_t, V_{x,t}, y_t, V_{y,t})^T$, action a_t, et $(\widehat{\sigma}_l)_{l=1}^L$
$\mathbf{s}_{t+1} \leftarrow \mathbf{A}\,\mathbf{s}_t + \mathbf{G}\,\mathbf{w}_t$
Calcul du l'intervalle angulaire θ_{t+1} associé avec $\mathbf{s}_{t+1}$.
Calcul du l'intervalle angulaire l_t associé avec a_t.
Le paramètre α_{t+1} en fonction de la distance R_{t+1}
$\widehat{\alpha}_{t+1} \sim \mathcal{CN}\big(\alpha_{t+1}, \widehat{\sigma}_{l_t}^2\big)$
$\Lambda_t \leftarrow 2\,|\widehat{\alpha}_{t+1}|^2/\widehat{\sigma}_{l_t}^2$
if $l_t \neq \theta_{t+1}$ **then**
  $o_{t+1} \leftarrow \emptyset$
else if $l_t = \theta_{t+1}$ **then**
  if $\Lambda_t \geq \lambda$ **then**
    $o_{t+1} \leftarrow |\widehat{\alpha}_{t+1}|$
  else
    $o_{t+1} \leftarrow \emptyset$
  end if
end if
$r_t \leftarrow \mathbf{1}\{\, l_t = \theta_{t+1}\}$
return $(\mathbf{s}_{t+1}, o_{t+1}, r_t)$

$$c_n = \sum\nolimits_{i=1}^{p} \rho_i c_{n-i} + w_n, \quad n \in (-\infty, +\infty), \quad (9)$$

Ce processus est piloté par des innovations w_n identiquement et indépendamment distribuées selon une loi t, avec une fonction de densité de probabilité p_w définie par :

$$p_w(w_n) = \frac{\mu}{\sigma_w^2 \pi} \left(\frac{\mu}{\xi}\right)^{\mu} \left(\frac{\mu}{\xi} + \frac{|w_n|^2}{\sigma_w^2}\right)^{-(\mu+1)}, \quad (10)$$

où $\mu \in (1, +\infty)$ est le paramètre de forme contrôlant la non-gaussianité de w_n, et le paramètre d'échelle est défini par $\xi = \frac{\mu}{\sigma_w^2(\mu-1)}$.

Dans les simulations, les paramètres utilisés sont : $p = 6$ pour l'ordre du processus AR, $\mu = 2$, $\sigma_w^2 = 1$, et le vecteur de coefficients $\boldsymbol{\rho}$ est défini comme suit :

$$\boldsymbol{\rho} = [0{,}5e^{-j2\pi\cdot 0{,}4},\ 0{,}6e^{-j2\pi\cdot 0{,}2},\ 0{,}7e^{-j2\pi\cdot 0},\ 0{,}4e^{-j2\pi\cdot 0{,}1},\ 0{,}5e^{-j2\pi\cdot 0{,}3},\ 0{,}6e^{-j2\pi\cdot 0{,}35}]^T. \quad (11)$$

Le nombre de canaux spatiaux $N = N_T N_R = 10^4$, le nombre d'intervalles d'angle $L = N_T = 100$, la puissance totale $P_T = 1$ et la probabilité de fausse alarme $P_{FA} = 10^{-4}$. Le nombre de simulations $N_{\text{sim}} = 10^4$ et de particules $N_p = 10^4$. Le paramètre est fixé à $\gamma = 0.8$ et l'horizon est de $T_{\text{max}} = 100$ étapes temporelles. Le paramètre UCB1, $c = \sqrt{2}$ équilibre entre l'exploration et exploitation, tandis que la profondeur de l'arbre est limitée à 2.
La moyenne des résultats est calculée sur 250 simulations de Monte Carlo. Afin d'établir une limite inférieure pour l'évaluation des performances, nous définissons un algorithme oracle. L'oracle est un algorithme idéalisé qui connaît le futur intervalle d'angle contenant la cible (mais pas ses coordonnées exactes ni sa vitesse) et qui prend toujours l'action optimale sur la base de cette connaissance. Il reçoit une observation et construit un état de croyance. Par essence, l'oracle est l'équivalent d'un POMCP sans faille. En outre, nous définissons

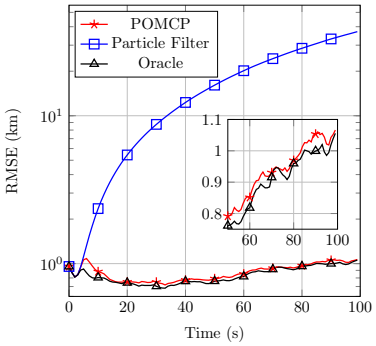

FIGURE 1 : La performance en RMSE (km) des algorithmes.

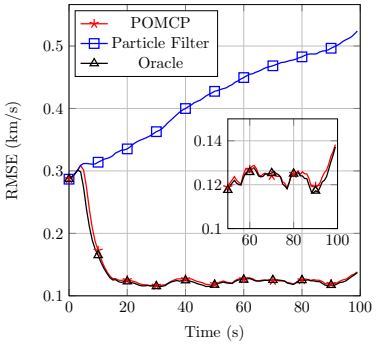

FIGURE 2 : La performance en RMSE (km/s) des algorithmes.

également l'algorithme avec le filtre à particule suivant : à l'instant t, une approximation de $b(.|h_t)$ est définie par l'ensemble B_t. Le filtre à particules doit anticiper l'état caché futur de la cible, c'est-à-dire calculer $\mathbb{E}(\mathbf{s}_{t+1}|h_t)$, de la même manière que dans [2].

$$\mathbb{E}(\mathbf{s}_{t+1}|h_t) \approx \frac{1}{|B_t|} \sum_{\mathbf{s} \in B_t} \mathbb{E}(\mathbf{s}_{t+1}|\mathbf{s}_t = \mathbf{s}) \qquad (12)$$

Le POMCP et le filtre à particules utiliseront le générateur de l'algorithme 1 pour effectuer des simulations et alimenter les ensembles de croyances à chaque itération.
On considère un scénario dont la cible est initialisée avec $\mathbf{s}_0 = (60 \text{ km}, 0.2 \text{ km/s}, -60 \text{ km}, 0.2 \text{ km/s})^T$, avec un bruit $\sigma_s = 0.03$. En moyenne, la cible est associé avec un SNR qui commence à -17 dB et diminue à -18 dB après 100 étapes temporelles.

Les figures 1 et 2 démontrent la nette supériorité de l'algorithme POMCP (rouge) face au filtre à particules (bleu) pour

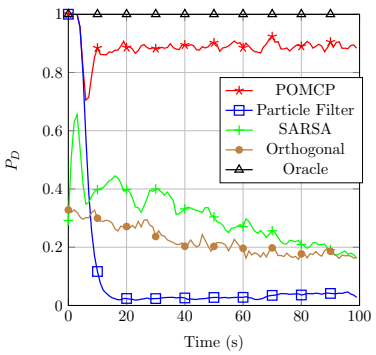

FIGURE 3 : La performance en P_D des algorithmes.

l'estimation des positions (km) et vitesses (km/s). Le POMCP maintient des erreurs RMSE proches de l'Oracle (noir), tandis que les erreurs du filtre à particules augmentent continuellement avec le temps. La figure 3 montre que le POMCP parvient à détecter une cible en mouvement rapide et maintient une probabilité de détection supérieure à 0.8, alors que le SARSA [1] rencontre des difficultés pour maintenir la détection.

7 Conclusion

Un algorithme POMCP original est proposé pour la détection et le suivi simultanés d'une cible mobile avec les systèmes radar MMIMO, même en présence de perturbations inconnues. Les travaux futurs étudieront l'impact des hyperparamètres du POMCP et étendront ce cadre aux scénarios multi-cibles.

Références

[1] Aya Mostafa AHMED, Alaa Alameer AHMAD, Stefano FORTUNATI, Aydin SEZGIN, Maria Sabrina GRECO et Fulvio GINI : A Reinforcement Learning Based Approach for Multitarget Detection in Massive MIMO Radar. *IEEE Transactions on Aerospace and Electronic Systems*, 57(5): 2622–2636, 2021.

[2] Imad BOUHOU, Stefano FORTUNATI, Leila GHARSALLI et Alexandre RENAUX : Pomdp-driven cognitive massive mimo radar : Joint target detection-tracking in unknown disturbances. *IEEE Transactions on Radar Systems*, 3:539–548, 2025.

[3] Stefano FORTUNATI, Luca SANGUINETTI, Fulvio GINI, Maria Sabrina GRECO et Braham HIMED : Massive MIMO Radar for Target Detection. *IEEE Transactions on Signal Processing*, 68:859–871, 2020.

[4] S. HAYKIN : Cognitive Radar : a Way of the Future. *IEEE Signal Processing Magazine*, 23(1):30–40, 2006.

[5] Leslie Pack KAELBLING, Michael L. LITTMAN et Anthony R. CASSANDRA : Planning and Acting in Partially Observable Stochastic Domains. *Artif. Intell.*, 101(1–2): 99–134, mai 1998.

[6] Levente KOCSIS et Csaba SZEPESVÁRI : Bandit Based Monte-Carlo Planning. *In* Johannes FÜRNKRANZ, Tobias SCHEFFER et Myra SPILIOPOULOU, éditeurs : *Machine Learning : ECML 2006*, pages 282–293, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[7] Brian W. RYBICKI et Jill K. NELSON : A Cognitive Tracking Radar using Continuous Space Monte Carlo Tree Search. *In 2022 IEEE Radar Conference (RadarConf22)*, pages 1–6, 2022.

[8] David SILVER et Joel VENESS : Monte-Carlo Planning in Large POMDPs. *In Proceedings of the 23rd International Conference on Neural Information Processing Systems - Volume 2*, NIPS'10, page 2164–2172, Red Hook, NY, USA, 2010. Curran Associates Inc.