

# LOCALISATION ET CARTOGRAPHIE SIMULTANÉES EN ROBOTIQUE (SLAM)

Depuis les années 1990, le SLAM (Simultaneous localization and mapping  $\equiv$  Localisation et cartographie simultanées) est sans doute l'un des problèmes de Robotique ayant donné lieu au plus grand nombre de contributions. Considérant un robot en mouvement dans un environnement inconnu (sans connaissance a priori), il consiste à construire un modèle du monde, et, concomitamment, à localiser le robot dans celui-ci. On obtient alors une carte métrique, pouvant être exploitée à des fins de navigation ou pour d'autres objectifs.

Le SLAM combine des données issues de capteurs proprioceptifs, qui délivrent une information relative au déplacement du robot (codeurs de roues, etc.) ainsi que des capteurs extéroceptifs, qui renseignent le robot sur son environnement (vision, télémètre laser, etc.). La version la plus élémentaire de l'algorithme exploite des primitives géométriques de l'environnement fixes et identifiables sans ambiguïté entre deux perceptions. Reprenant une terminologie utilisée en navigation maritime, ces primitives sont appelées « amers ». Lorsque le SLAM est basé sur des séquences d'images, ces amers peuvent être des cadres de portes, des posters quadrangulaires, etc.

De nombreuses approches du SLAM ont reposé sur le filtrage Bayésien : extensions non linéaires du filtrage de Kalman, filtre information, filtrage particulaire, etc. Le vecteur d'état  $\mathbf{x}$  à estimer est naturellement constitué de la situation  $\mathbf{r}$  du robot et des situations  $\mathbf{m}_1, \dots, \mathbf{m}_M$  des  $M$  primitives du modèle, ces situations étant exprimées dans un repère lié au monde. La dynamique a priori de  $\mathbf{r}$  est souvent basée sur l'odométrie du robot, *i.e.*, sur la reconstitution de son mouvement global à partir de son modèle cinématique et des déplacements (connus) des roues. Le sous-ensemble  $\mathbf{m}_1, \dots, \mathbf{m}_M$  relatif aux amers n'évolue pas au fil du temps, ceux-ci étant fixes. Le vecteur des observations  $\mathbf{z}$  assimilées par le filtre est constitué des données brutes perçues. Ces mesures doivent être appariées avec leur contrepartie dans  $\mathbf{m}_1, \dots, \mathbf{m}_M$ , *i.e.* telle perception doit être associée à tel amer correspondant à tel sous-ensemble de composantes de  $\mathbf{x}$ . Ce problème d'*association de données* peut être significativement simplifié (voire éliminé) si les amers sont convenablement sélectionnés.

On réalise alors itérativement le cycle [1. bouger le robot]  $\rightarrow$  [2. apparier les données perçues]  $\rightarrow$  [3. mettre à jour le modèle]. L'étape [1.] sert de base à la prédiction de  $\mathbf{x}$  entre deux instants successifs  $k-1$  et  $k$  préalablement au prélèvement d'une mesure  $\mathbf{z}_k = z_k$  à l'instant  $k$ ; l'étape [2.] consiste à associer  $z_k$  au sous-vecteur  $\mathbf{m}_m$  correspondant dans  $\mathbf{x}$ ,  $m \in \{1, \dots, M\}$ , en augmentant si besoin  $M$  si l'amer relatif à  $z_k$  ne figure pas déjà dans  $\mathbf{x}$ ; l'étape [3.] consiste à assimiler  $z_k$ . La dimension de  $\mathbf{x}$  croît donc au fil du temps tant qu'apparaissent de nouveaux amers. La dimension de  $\mathbf{z}$  peut ne pas être fixe.

En général, le problème est fortement non linéaire. On en propose toutefois ici une version extrêmement simplifiée. Ainsi, le robot est doté d'une locomotion omnidirectionnelle, et est assimilable à un point  $R$ . Il effectue un mouvement circulaire plan. L'environnement comporte deux amers  $A_1$  et  $A_2$ . Le robot mobile et les amers fixes sont repérés par leurs coordonnées  $(u^R, v^R)$  et  $(u^1, v^1), (u^2, v^2)$  relatives à un repère  $(O, \vec{x}, \vec{y})$ , où  $O$  désigne le centre de la trajectoire circulaire idéale que décrirait le robot en l'absence de perturbations sur sa dynamique (glissement, etc.).

Le capteur placé sur le robot (idéalisé, car n'ayant pas d'équivalent « physique ») délivre, à une erreur de mesure près, les composantes dans la base  $(\vec{x}, \vec{y})$  des vecteurs  $\overrightarrow{RA_1}$  et  $\overrightarrow{RA_2}$  qui séparent le robot des amers à condition que ces composantes soient positives. En d'autres termes, le capteur

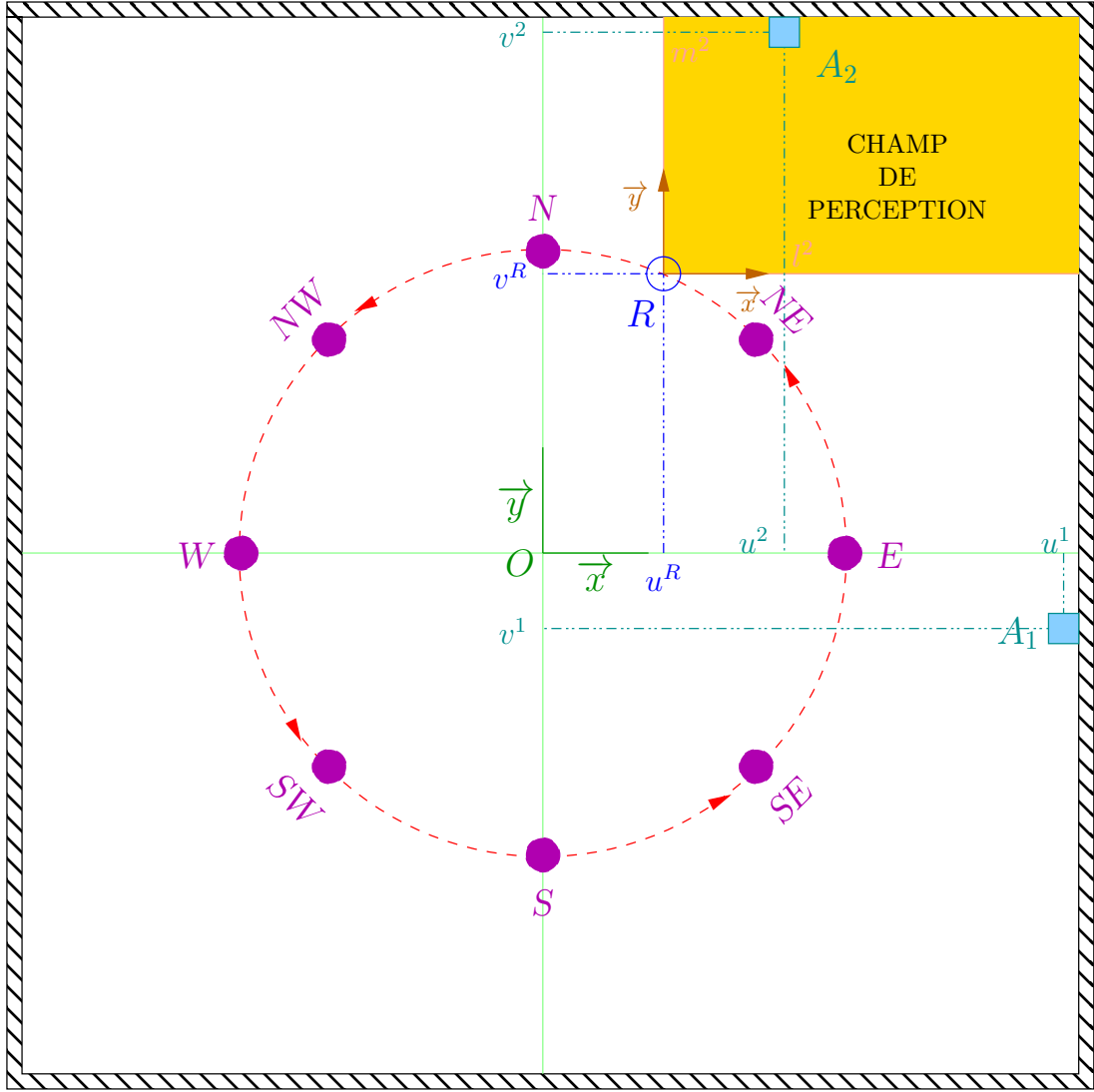


FIGURE 1 – Problème considéré.

délivre l'ensemble des composantes  $l_n = \overrightarrow{RA_m} \cdot \vec{x}$ ,  $m_n = \overrightarrow{RA_m} \cdot \vec{y}$  pour tout  $m \in \{1, 2\}$  tel que  $A_m$  soit « recouvert » par le premier quadrant délimité par  $(R, \vec{x}, \vec{y})$ . Notons que le repère  $(R, \vec{x}, \vec{y})$  est en translation circulaire mais conserve une orientation constante. Les mesures sont prélevées lorsque le robot se trouve dans l'une des huit zones *Est*, *Nord-Est*, *Nord*, *Nord-Ouest*, *Ouest*, *Sud-Ouest*, *Sud*, *Sud-Est*, respectivement désignées par *E*, *NE*, *N*, *NW*, *W*, *SW*, *S*, *SE*. Le robot a en permanence connaissance du nombre d'amers situés dans le champ de perception de son capteur. Le problème est résumé sur la Figure 1.

### Notations et Données

On désigne par  $\mathbb{I}$  toute matrice identité et par  $\mathbb{O}$  toute matrice constituée de zéros. Les dimensions de ces matrices sont sous-entendues lorsqu'elles découlent du contexte. La notation  $\text{blkdiag}(A_1, \dots, A_L)$  consiste en la matrice bloc-diagonale constituée de l'assemblage de  $A_1, \dots, A_L$ . L'opérateur de transposition est noté  $'$ . Enfin,  $\mathbf{x} \sim \mathcal{N}(\bar{\mathbf{x}}, P)$  signifie que la variable aléatoire (a priori multidimensionnelle)  $\mathbf{x}$  suit une loi normale de moyenne  $\bar{\mathbf{x}}$  et de covariance  $P$ . On notera éventuellement  $\mathcal{N}(\mathbf{x}; \bar{\mathbf{x}}, P)$  la densité de probabilité  $p_{\mathbf{x}}(\mathbf{x})$  associée.

Soient  $\mathbf{r} = (u^R, v^R)'$ ,  $\mathbf{m}_1 = (u^1, v^1)'$ ,  $\mathbf{m}_2 = (u^2, v^2)'$ , et  $\mathbf{x} = ((\mathbf{r})', (\mathbf{m}_1)', (\mathbf{m}_2)') \in \mathbb{R}^6$  le vecteur d'état (caché) complet. Les valeurs de ces quantités à l'instant  $k$  sont respectivement désignées par  $\mathbf{r}_k$ ,  $u_k^R$ ,  $v_k^R$ ,  $\mathbf{m}_{1,k}$ ,  $u_k^1$ ,  $v_k^1$ ,  $\mathbf{m}_{2,k}$ ,  $u_k^2$ ,  $v_k^2$ ,  $\mathbf{x}_k$ .

Dans l'ensemble du sujet, on note

$$\mathbf{r}_{k+1} = F^R \mathbf{r}_k, \text{ avec } F^R = \begin{pmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix}, \quad (1)$$

l'équation d'état qui unirait les valeurs de  $\mathbf{r}$  à deux instants successifs (*i.e.*, passage  $E \rightarrow NE \rightarrow N \rightarrow NW \rightarrow W \rightarrow SW \rightarrow S \rightarrow SE$ ) si la trajectoire du robot était parfaitement circulaire. Les amers sont supposés admettre une dynamique nulle, et satisfaire l'équation

$$\forall m \in \{1, 2\}, \mathbf{m}_{m,k+1} = \mathbf{m}_{m,k}. \quad (2)$$

Du fait de phénomènes inévitables (glissement, etc.) qui font que le robot ne suit pas parfaitement la trajectoire idéale décrite par (1), le vecteur d'état complet  $\mathbf{x} = ((\mathbf{r})', (\mathbf{m}_1)', (\mathbf{m}_2)')$  obéit à une équation d'état stochastique à temps discret de la forme

$$\mathbf{x}_{k+1} = F \mathbf{x}_k + \mathbf{w}_k, \quad (3)$$

où le bruit de dynamique  $\mathbf{w}$  est blanc stationnaire et vérifie  $\forall k, \mathbf{w}_k \sim \mathcal{N}(0, Q_w)$ . La matrice de dynamique  $F$  et la matrice de covariance de bruit  $Q_w$  (données) admettent donc une structure bloc-diagonale, *i.e.*,  $F = \text{blkdiag}(F^R, \mathbb{I}, \mathbb{I})$  et  $Q_w = \text{blkdiag}(Q_w^R, \mathbb{O}, \mathbb{O})$ .

Le robot part de la zone  $E$ , de sorte que le vecteur aléatoire d'état à l'instant initial  $k = 0$  vérifie

$$\mathbf{x}_0 \sim \mathcal{N}(m_{x_0}, P_0), \quad (4)$$

avec  $m_{x_0}$  et  $P_0$  donnés. Pour tout  $k$ ,  $\mathbf{x}_0$  et  $\mathbf{w}_k$  sont supposés mutuellement indépendants.

Soient  $l_k^1$ ,  $m_k^1$ ,  $l_k^2$ ,  $m_k^2$  les mesures (à valeurs réelles) délivrées par le capteur à tout instant  $k \geq 1$ , et  $\mathbf{z}_k$  le vecteur regroupant celles-ci. Selon les valeurs de  $k$ , la visibilité des amers change, de sorte que la dimension de  $\mathbf{z}_k$  varie dans  $\{0, 2, 4\}$ . L'équation de mesure stochastique qui unit  $\mathbf{z}$  et  $\mathbf{x}$  s'écrit

$$\mathbf{z}_k = H_k \mathbf{x}_k + \mathbf{v}_k, \quad (5)$$

avec  $\mathbf{v}$  un bruit de mesure blanc Gaussien stationnaire, tel que  $\mathbf{v}_k \sim \mathcal{N}(0, R_v)$ . Pour tous  $k, k'$ ,  $\mathbf{x}_0$ ,  $\mathbf{w}_k$ ,  $\mathbf{v}_{k'}$  sont supposés mutuellement indépendants. Les dimensions des matrices (données) d'observation  $H_k$  et de covariance  $R_v$  du bruit varient éventuellement en chaque instant  $k$  avec la dimension de  $\mathbf{z}_k$ .

On fournit la fonction MATLAB `simulationDonnees.m`, dont le prototype est décrit Figure 2. Cette fonction permet de simuler une expérience. Par conséquent, elle fournit un résultat différent lors de chacun de ses appels. Outre le nombre d'instants  $N$ , l'horizon temporel  $T$ , les trajectoires temporelles  $\mathbf{Z}$  et  $\mathbf{X}$  des processus de mesure et d'état  $\mathbf{z}$  et  $\mathbf{x}$  pour l'expérience considérée, elle fournit également la matrice  $\mathbf{F}$  ( $F$ ), la matrice `Hfull` (*i.e.*,  $H$  lorsque les deux amers sont visibles), ainsi que les statistiques `mX0, PX0, Qw, Rv` (respectivement  $m_{X_0}$ ,  $P_0$ ,  $Q_w$ ,  $R_v$ ). Il convient de préciser que si l'expérience était menée sur un robot réel, seuls  $N, T, \mathbf{Z}$  seraient accessibles (on pourrait toutefois obtenir  $\mathbf{X}$  *via* un système de capture de mouvement par exemple).

On fournit également la fonction `ellipse`, qui, selon le prototype reporté Figure 3, permet de tracer l'ensemble de volume minimal dans lequel se réalise une variable aléatoire Gaussienne à deux dimensions.

```
function [N,T,Z,F,Hfull,mX0,PX0,Qw,Rv,X] = simulationDonnees(plot_p);
%SIMULATIONDONNEES
% Simulation d'une expérience : déplacement du robot, recueil des mesures, etc.
% Syntaxe d'appel : [N,T,Z,F,Hfull,mX0,PX0,Qw,Rv,X] = simulationDonnees(plot_p);
%
% Entrée :
% . plot_p : si égal à 1, produit un affichage animé, sinon n'affiche rien
%
% Sorties :
% . N : nombre d'échantillons temporels
% . T : vecteur des instants d'échantillonnage (1xN)
% . Z : réalisation du processus aléatoire de mesure (4xN, éventuellement avec composantes de type NaN)
% -> et comme ceci est de la simulation, sont également accessibles
% . F, Hfull : matrices du modèle (respectivement (6x6), (4x6))
% . mX0 : espérance du vecteur d'état à l'instant 0 (6x1)
% . PX0 : covariance du vecteur d'état à l'instant 0 (6x6)
% . Qw : covariance du bruit de dynamique (supposé stationnaire) (6x6)
% . Rv : covariance du bruit de mesure (supposé stationnaire) (4x4)
% . X : réalisation du processus aléatoire d'état (6xN)
%
% Z et X admettent AUTANT DE COLONNES QUE D'INSTANTS
```

FIGURE 2 – Prototype de `simulationDonnees.m`

```
function h = ellipse(mx,Px,color)
%ELLIPSE
% Trace l'ellipse contenant 99% des réalisations d'une variable aléatoire
% Gaussienne 2D de moyenne mx et de covariance Px.
```

FIGURE 3 – Prototype de `ellipse.m`

## Travail demandé

### Simulation – Écriture du modèle

1. Exécuter un “run” de la fonction `simulationDonnees`.
  - (a) Expliquer les informations fournies par l'animation graphique.
  - (b) Expliquer le contenu des variables `T,Z,X`.
  - (c) Donner un sens *physique* à `mX0,P0,Qw,Rv`.
  - (d) Rappeler la raison pour laquelle le filtre de Kalman permet une résolution exacte du problème de l'estimation Bayésienne du vecteur d'état caché à tout instant sur la base des observations disponibles.
  - (e) Implémenter cette solution exacte. *Si vous n'avez pas traité cette question en 2ASRI, vous êtes encouragé-e à lui consacrer un certain temps en sus des séances propres à ce TP...*
2. Établir l'expression de
  - (a) la densité de probabilité de l'état initial (ou « loi initiale »)  $p(x_0)$ ;
  - (b) la loi de dynamique a priori  $p(x_k|x_{k-1})$ ; *si besoin, on utilisera une version légèrement modifiée de cette loi, dans laquelle on supposera que la position (inconnue) des amers est constante en moyenne mais sujette à des petites variations aléatoires de faible dispersion*;
  - (c) le modèle d'observation  $p(z_k|x_k)$  (selon la visibilité des amers).

### Implémentation d'un filtre particulaire SIS

3. Écrire l'algorithme du filtre particulaire "Sequential Importance Sampling" (SIS) pour le problème considéré. La fonction d'importance sera sélectionnée égale à la loi de dynamique a priori, éventuellement en supposant que la position (inconnue) des amers est constante en moyenne mais sujette à des petites variations aléatoires de faible dispersion.
4. Procéder à son implémentation pour l'initialisation ainsi que les modèles de bruits identiques à ceux exploités dans `simulationDonnees`. Exécuter le filtre sur la base d'une réalisation du processus de mesure.
  - (a) Dédurre les estimés des moyennes et covariances a posteriori  $\hat{x}_{k|k}$  et  $P_{k|k}$  du vecteur d'état  $\mathbf{x}_k$ , conditionnellement à  $\mathbf{z}_{1:k} = \mathbf{z}_{1:k}$ . Observer l'évolution de ces quantités.
  - (b) Sur cette base, établir et tracer des domaines de volume minimal dans lesquels les réalisations des vecteurs  $\mathbf{r}_k$ ,  $\mathbf{m}_{1,k}$  et  $\mathbf{m}_{2,k}$  devraient idéalement se trouver avec une certaine probabilité. Par exemple, (et ceci est d'autant plus vrai qu'on se situe dans un contexte linéaire Gaussien),
    - à chaque instant  $k$ , sur la base de  $\hat{x}_{k|k}$  et  $P_{k|k}$ , définir pour chaque composante du vecteur d'état (caché)  $\mathbf{x}_k$ , un « couloir » dans lequel elle doit se réaliser avec une probabilité théorique de 99 % ;
    - (mieux !) à chaque instant  $k$ , sur la base de  $\hat{x}_{k|k}$  et  $P_{k|k}$ , définir les ellipses de confiance dans lesquels doivent se réaliser  $\mathbf{r}_k$ ,  $\mathbf{m}_1$ ,  $\mathbf{m}_2$  avec une probabilité théorique de 99 % (fonction `ellipse`).
5. Tester l'algorithme pour divers choix du nombre de particules. Quelles difficultés observe-t-on ? Vérifier la dégénérescence du nuage par calcul de la taille efficace du  $N$ -échantillon.

### Filtre particulaire SIR

6. Compléter l'algorithme SIS précédent par une étape de rééchantillonnage du nuage, lorsque le nombre de particules efficaces se situe en deçà d'un certain seuil à définir, de façon à obtenir un algorithme de "Sampling Importance Resampling" (SIR).
7. Tester ce nouvel algorithme. Observer le phénomène de « fermeture de boucle », où l'observation d'un amer déjà vu dans le passé permet, grâce à des corrélations établies a posteriori par le filtre entre les composantes de  $\mathbf{x}$ , d'apporter des informations sur la situation d'autres amers.
8. Éventuellement, comparer les valeurs *exactes* des moments  $\hat{x}_{k|k}$  et  $P_{k|k}$  de la loi de filtrage calculées par le filtre de Kalman avec leurs estimés délivrés par le filtre particulaire sur plusieurs "runs" de celui-ci, pour diverses valeurs du nombre de particules.
9. Proposer une définition alternative de la fonction d'importance, permettant un échantillonnage des particules prenant à la fois en compte la dynamique a priori du robot et une information apportée par les mesures.

### Stratégie élaborée

10. Implémenter une version Rao-Blackwellisée du filtre, telle que : la position du robot est estimée au moyen d'une loi ponctuelle ; à chaque particule entrant dans la constitution de cette loi marginale est associée la loi (Gaussienne) conditionnelle de la position de chaque amer. Tester.