



Filtrage Bayésien et Approximation Particulaire

Recalage altimétrique de navigation inertielle

Pirashanth RATNAMOGAN
Imad EL HANAFI
Promotion 2018

Professeur :
M. François LE GLAND

29 Septembre 2017

Table des matières

1	Introduction et Problématique	4
1.1	Mesures intertielles et modélisation	4
1.2	Modèle d'état	6
1.3	Relation entre observation et état $(\delta r_k, \delta v_k)$	6
2	Amélioration du résultat grâce à l'algorithme SIR	8
2.1	Calcul de la vraisemblance	8
2.2	L'algorithme SIR avec redistribution multinomiale à chaque pas de temps	9
2.2.1	L'implémentation de l'algorithme	9
2.3	Résultats	10
2.3.1	Impact du nombre de particules N	10
2.3.2	Suivi du nuage pour un N fixé	11
3	Algorithme hybride SIR/SIS	12
3.1	L'implémentation du SIR/SIS	12
3.2	Résultats SIS seul	13
3.3	Résultats hybride SIS/SIR	14
4	Conclusion	16

1 Introduction et Problématique

Le but de ce premier TP du cours de Filtrage Bayésien est de retrouver, à l'aide des outils présentés au cours, la trajectoire d'un avion. Dans ce but on dispose d'une mesure inertielle bruitée (accélération) et de la mesure du relief directement sous l'avion. L'avion survole une zone dont le relief est donné et connu par $h(r)$ en chaque point de cordonnée $r \in [x_{min}, x_{max}] \times [y_{min}, y_{max}]$. L'avion est repéré par la position horizontale r et l'altitude z . On suppose que l'avion se déplace à l'altitude $z = z_0$ constante et à vitesse horizontale constante $v = v_0$.

La carte numérique du relief `mnt.data` (modèle numérique du terrian) ainsi que la trajectoire (position exacte de l'avion) `traj.mat` sont fournies. Notre but sera de trouver un estimateur performante de cette trajectoire en utilisant uniquement les différentes mesures que l'on a à disposition. La figure ci-dessous montre la trajectoire exacte (en rouge) $r_k = (r_{1,k}, r_{2,k})$ pour k entre l'instant 0 et T , et la carte numérique du terrain :

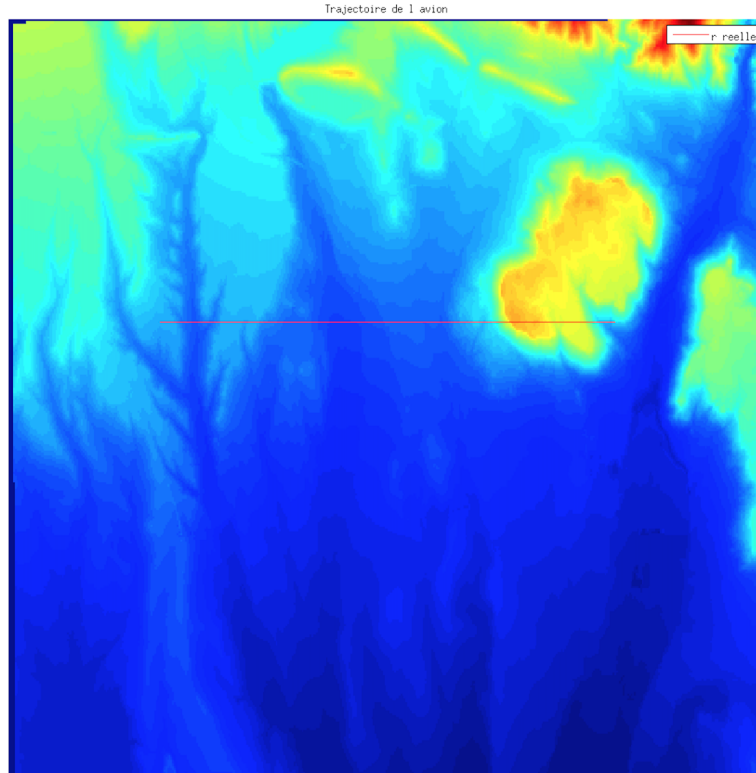


Figure 1: Modèle numérique de terrain et trajectoire exacte

1.1 Mesures intertielles et modélisation

Pour permettre à l'avion d'estimer sa propre position horizontale r_k (x et y) et vitesse v_k à chaque instant $t_k \in [0, T]$, on mesure à l'aide d'accéléromètres et de

gyroscopes l'accélération horizontale a_k^{INS} avec un pas de temps $\Delta = t_k - t_{k-1} = cte$. L'erreur de mesure est modélisée par un bruit gaussien : $w_k^{INS} \sim \mathcal{N}(0, \sigma_{INS}^2 I_2)$.

$$a_k^{INS} = a_k + w_k^{INS}$$

pour $k \in \{1, \dots, T = nmax - 1\}$ avec a_k l'accélération horizontale exacte de l'avion à l'instant k .

L'intégration de cette équation, sachant les conditions initiales (r_0, v_0) nous permet d'obtenir l'équation satisfaite par (r_k^{INS}, v_k^{INS}) qui représentent une estimation de la position et la vitesse horizontale.

Ceci se traduit par $r_k^{INS} = r_{k-1}^{INS} + \Delta v_{k-1}^{INS}$ et $v_k^{INS} = v_{k-1}^{INS} + \Delta a_{k-1}^{INS}$. On a donc :

$$\begin{pmatrix} r_k^{INS} \\ v_k^{INS} \end{pmatrix} = \begin{pmatrix} I_2 & \Delta I_2 \\ 0 & I_2 \end{pmatrix} \begin{pmatrix} r_{k-1}^{INS} \\ v_{k-1}^{INS} \end{pmatrix} + \Delta \begin{pmatrix} 0 \\ a_k^{INS} \end{pmatrix} \quad (1.1)$$

avec $\begin{pmatrix} r_1^{INS} \\ v_1^{INS} \end{pmatrix} = \begin{pmatrix} r_0 \\ v_0 \end{pmatrix}$

Après avoir récupéré les données a_k^{INS} (`ins.mat`) et en utilisant l'équation précédente, on obtient les valeurs r_k^{INS} et v_k^{INS} de façon récursive.

La figure ci-dessous montre pour chaque instant k entre 1 et T la position exacte r_k et son estimation inertielle r_k^{INS} .

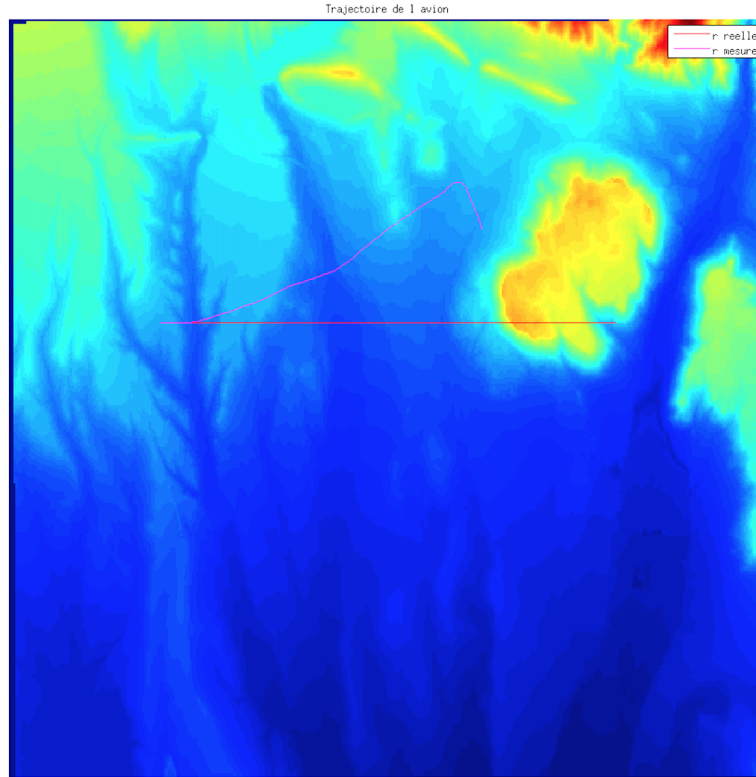


Figure 2: Modèle numérique de terrain, trajectoire exacte et trajectoire inertielle

Remarque : On constate que l'estimation basée sur la mesure inertielle est très différente de la trajectoire réelle. Cela est dû à l'erreur de mesure (modélisée par la variable aléatoire w_k^{INS}). A partir de l'équation (1.1) on remarque qu'il y a un cumul d'erreur dans l'estimateur basé sur la mesure inertielle ce qui explique la déviation observée dans *Figure 2*.

1.2 Modèle d'état

On va s'intéresser aux nouvelles variables d'état qui représentent l'erreur d'estimation inertielle sur la position et la vitesse : $\delta r_k = r_k - r_k^{INS}$ et $\delta v_k = v_k - v_k^{INS}$

On sait que pour $k \in \{1, \dots, T\}$:

$$\begin{cases} r_k^{INS} = r_{k-1}^{INS} + \Delta v_{k-1}^{INS} \\ v_k^{INS} = v_{k-1}^{INS} + \Delta a_{k-1}^{INS} \end{cases} \quad \begin{cases} r_k = r_{k-1} + \Delta v_{k-1} \\ v_k = v_{k-1} + \Delta a_{k-1} \end{cases} \quad (1.2)$$

avec $r_1^{INS} = r_0$, $v_1^{INS} = v_0$ et $r_1 = r_0$, $v_1 = v_0$

Donc

$$\begin{cases} \delta r_k = r_k - r_k^{INS} = (r_{k-1} + \Delta v_{k-1}) - (r_{k-1}^{INS} + \Delta v_{k-1}^{INS}) \\ \delta v_k = v_k - v_k^{INS} = (v_{k-1} + \Delta a_{k-1}) - (v_{k-1}^{INS} + \Delta a_{k-1}^{INS}) \end{cases} \quad (1.3)$$

Sachant que $a_k^{INS} = a_k + w_k^{INS}$, on obtient :

$$\begin{cases} \delta r_k = \delta r_{k-1} + \Delta \delta v_{k-1} \\ \delta v_k = \delta v_{k-1} + \Delta (w_k^{INS}) \end{cases} \quad (1.4)$$

avec $\delta r_1 = 0$ et $\delta v_1 = 0$

D'où l'équation d'état :

$$\begin{pmatrix} \delta r_k \\ \delta v_k \end{pmatrix} = \begin{pmatrix} I_2 & \Delta I_2 \\ 0 & I_2 \end{pmatrix} \begin{pmatrix} \delta r_{k-1} \\ \delta v_{k-1} \end{pmatrix} + \Delta \begin{pmatrix} 0 \\ w_k^{INS} \end{pmatrix} \quad (1.5)$$

avec $\begin{pmatrix} \delta r_1 \\ \delta v_1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$ et $w_k^{INS} \sim \mathcal{N}(0, \sigma_{INS}^2 I_2)$

1.3 Relation entre observation et état $(\delta r_k, \delta v_k)$

On recueille au moyen d'un radar altimétrique avec le même pas de temps Δ une mesure d_k^{ALT} de la hauteur de l'avion au-dessus du relief, avec une erreur modélisée par un bruit gaussien centré de variance σ_{ALT}^2 . De plus, à l'aide d'un baromètre on mesure z_k^{BAR} l'altitude de l'avion par rapport au niveau de la mer (niveau zéro) avec un bruit gaussien centré de variance σ_{BAR}^2 .

On note par $h(r_k)$ la hauteur au point de coordonnée horizontale r_k . On a la relation suivante

$$\begin{cases} d_k^{ALT} = (z_k - h(r_{k-1})) + w_k^{ALT} \\ z_k^{BAR} = z_k + w_k^{BAR} \end{cases} \quad (1.6)$$

Donc la hauteur du relief survolé à l'instant t_k à partir des mesures est donnée par la relation :

$$h_k^{ALT} = z_k^{BAR} - d_k^{ALT} = h(r_k) + w_k^{BAR} - w_k^{ALT} \quad (1.7)$$

Sachant que $\delta r_k = r_k - r_k^{INS}$, on obtient l'équation qui lie l'observation et les variables d'état cachées :

$$h_k^{ALT} = h(r_k^{INS} + \delta r_k) + w_k^{BAR} - w_k^{ALT} \quad (1.8)$$

La hauteur mesurée h_k^{ALT} est fournie dans `alt.mat`.

Pour représenter graphiquement la hauteur réelle du relief survolé $h(r_k)$ on doit retrouver les indices (i, j) associés au point r_k et ensuite on utilise la carte numérique `map` : $h(r_k) = \text{map}(i, j)$. On note par (N_1, N_2) la taille de `map`. Dans ce maillage, les indices (i, j) d'un point r_k se trouvent facilement à l'aide de la formule suivante :

$$(i, j) = (E[N_1 \frac{r_k(1) - x_{min}}{x_{max} - x_{min}}] + 1, E[N_2 \frac{r_k(2) - y_{min}}{y_{max} - y_{min}}] + 1) \quad (1.9)$$

$E[\alpha]$ indique la partie entière de α .

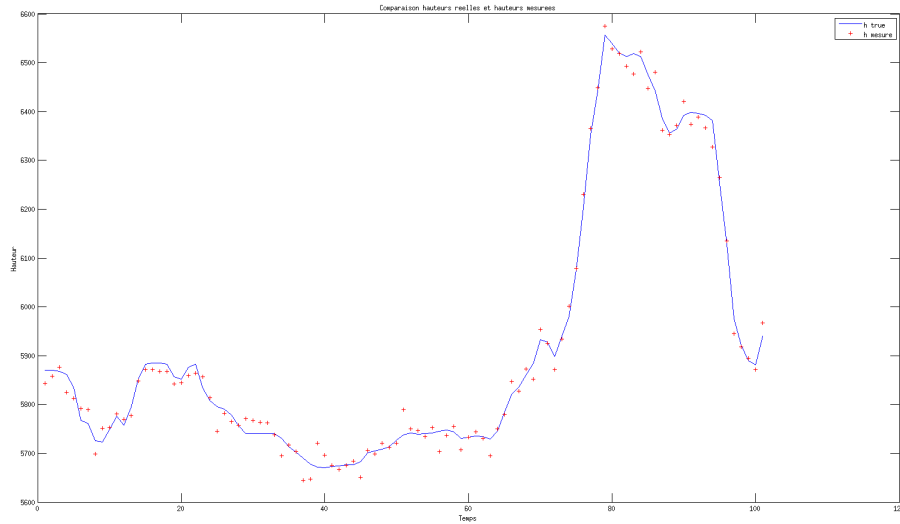


Figure 3: Profil réel du terrain survolé et mesures altimétriques

On voit directement que l'estimation de l'évolution de la hauteur du relief que l'on a est très proche du vrai relief sous l'avion. Le fait d'utiliser cette mesure de la hauteur du relief semble donc être une très bonne idée.

2 Amélioration du résultat grâce à l'algorithme SIR

L'algorithme SIR (sampling with importance resampling) est un algorithme qui permet de réaliser une simulation de la situation de l'évolution de la trajection de l'avion en se basant sur un calcul de vraisemblance. L'algorithme SIR a la spécificité de redistribuer les données en fonction de leurs vraisemblances ce qui permet à la simulation de rester proche des mesures effectuées dans la situation réelle. L'algorithme SIR a la particularité d'effectuer un "importance resampling" qui permet d'éliminer les trajectoires aberrantes à chaque pas de temps, la prédiction aux temps suivants se basent donc sur des trajectoires plausibles.

2.1 Calcul de la vraisemblance

Problématique : On cherche le meilleur écart δr à la mesure inertielle r^{INS} , qui permet à la hauteur mesurée h^{ALT} d'être proche de la hauteur connue du relief à la position estimée $h(r^{INS} + \delta r)$

Etant donné que la loi de h_k^{ALT} , notre mesure, est donnée pour tout $k \in \{1, \dots, T+1\}$ par :

$$h_k^{ALT} = h(r_k^{INS} + \delta r_k) + w_k^{BAR} - w_k^{ALT} \quad (2.1)$$

$$w_k^{BAR} \sim \mathcal{N}(0, \sigma_{BAR}^2), w_k^{ALT} \sim \mathcal{N}(0, \sigma_{ALT}^2) \quad (2.2)$$

On a donc, par somme de deux lois gaussiennes centrées :

$$w_k^{BAR} - w_k^{ALT} \sim \mathcal{N}(0, \sigma_{BAR}^2 + \sigma_{ALT}^2) \quad (2.3)$$

La vraisemblance au temps k se déduit donc facilement puisqu'on a que :

$$h_k^{ALT} - h(r_k^{INS} + \delta r_k) \sim \mathcal{N}(0, \sigma_{BAR}^2 + \sigma_{ALT}^2) \quad (2.4)$$

On note q_k la fonction vraisemblance au temps k , celle ci est donc donnée par :

$$q_k(\delta r_k) \propto \exp\left(-\frac{(h_k^{ALT} - h(r_k^{INS} + \delta r_k))^2}{2(\sigma_{BAR}^2 + \sigma_{ALT}^2)}\right) \quad (2.5)$$

Par la suite on oublie la constante qui n'apporte rien à la comparaison des δr

2.2 L'algorithme SIR avec redistribution multinomiale à chaque pas de temps

2.2.1 L'implémentation de l'algorithme

Maintenant que l'on a la vraisemblance, on peut utiliser l'algorithme SIR avec redistribution multinomiale (qui nous permet de tirer des particules en fonction de leur plausabilité par rapport aux autres particules). L'algorithme utilisé dans notre cadre est détaillé plus bas .

On note **update-delta** : $(\delta r_{previous}) \mapsto \mathbf{update-delta}(\delta r_{previous})$ la fonction qui permet de faire avancer aléatoirement la particule $\delta r_{previous}$ d'un pas de temps suivant l'équation d'état 1.5

On note **vraisemblance** : $(\delta r, t) \mapsto \mathbf{vraisemblance}(\delta r, t)$ la fonction qui permet de calculer la vraisemblance de la particule δr au temps t par l'expression 2.5. On note **fct-multi** : $(\delta r, w, N) \mapsto \mathbf{fct-multi}((\delta r, w, N))$ la fonction qui permet de tirer N nouvelles particules issues de δr avec des probabilités proportionnelles aux poids w

On note N le nombre de particules utilisés, T le temps final de la mesure de position recherchée.

On **initialise** en tirant N particules initiales $(\xi_1^i)_{i \in \{1, \dots, N\}}$ suivant la loi

$$\xi_1 = \begin{pmatrix} \delta r_1 \\ \delta v_1 \end{pmatrix} \sim \mathcal{N}(0, \begin{pmatrix} \sigma r_0 I_2 & 0 \\ 0 & \sigma v_0 I_2 \end{pmatrix}) \quad (2.6)$$

On calcule $w_1 = (\mathbf{vraisemblance}((\xi_0^i), 1))_{i \in \{1, \dots, N\}}$

On calcule les poids $p_1 = (\frac{w_1^i}{\sum_{i \in \{1, \dots, N\}} w_1^i})_{i \in \{1, \dots, N\}}$

On stocke la valeur estimée $\bar{\xi}_1 = p_1^T \xi_1$

for $k \in \{2, \dots, T+1\}$ **do**

 On calcule : $\hat{\xi}_{k-1} = \mathbf{fct-multi}(\xi_{k-1}, w_{k-1}, N)$

 On calcule : $\xi_k = (\mathbf{update-delta}(\hat{\xi}_{k-1}^i))_{i \in \{1, \dots, N\}}$

 On calcule : $w_k = (\mathbf{vraisemblance}((\xi_k^i), 1))_{i \in \{1, \dots, N\}}$

 On calcule : $p_k = (\frac{w_k^i}{\sum_{i \in \{1, \dots, N\}} w_k^i})_{i \in \{1, \dots, N\}}$

 On stocke la valeur estimée : $\bar{\xi}_k = p_k^T \xi_k$

end

Le vecteur décrivant la trajectoire est donné par :

$$(r_1^{INS} + \bar{\xi}_1(1:2), \dots, r_k^{INS} + \bar{\xi}_k(1:2), \dots, r_{T+1}^{INS} + \bar{\xi}_{T+1}(1:2)) \quad (2.7)$$

Algorithm 1: Algorithme SIR qui permet d'améliorer notre estimation de la position de l'avion

2.3 Résultats

2.3.1 Impact du nombre de particules N

Ci-dessous on trouve les résultats de l'algorithme SIR pour différents nombres de particules utilisés à chaque pas de temps.

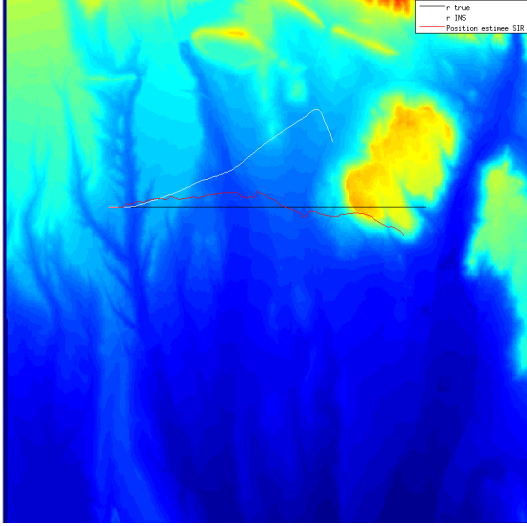


Figure 4: N=100

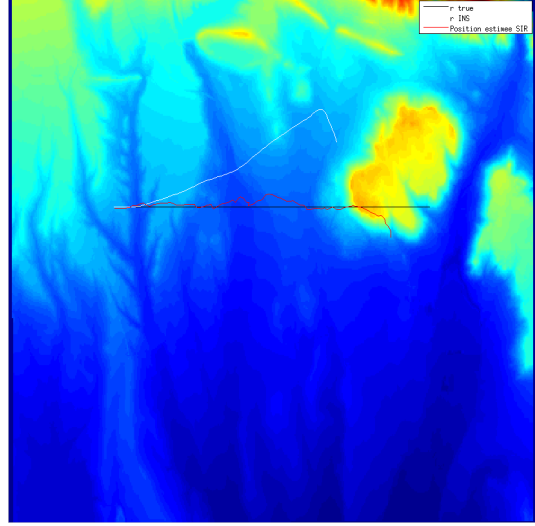


Figure 5: N=500

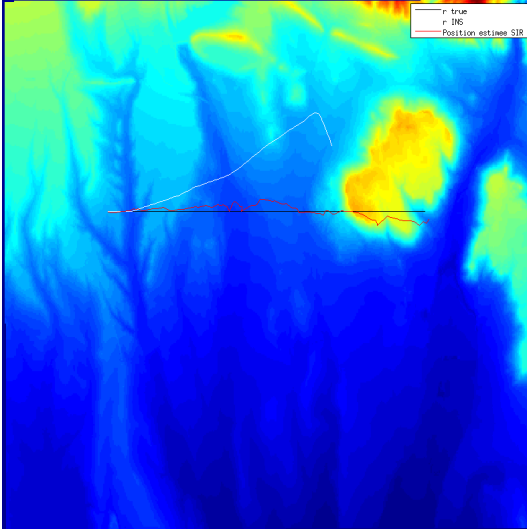


Figure 6: N=1000

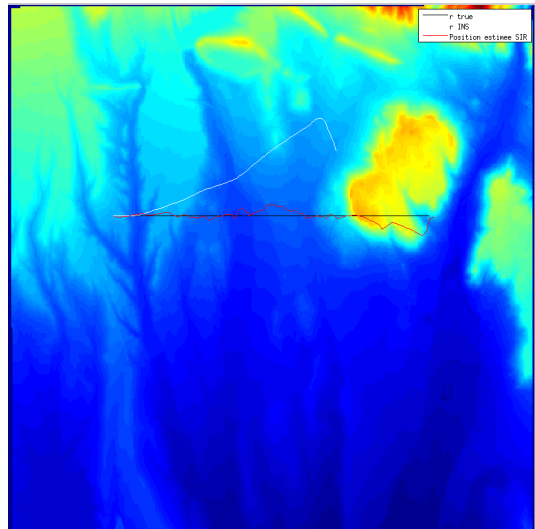


Figure 7: N=5000

Globalement on voit que le résultat basé sur la mesure inertielle seule est largement amélioré grâce à l'utilisation de l'algorithme SIR. Plus le nombre de particules utilisé est grand plus on se rapproche significativement de la trajectoire réelle. Cela s'explique facilement par la plus grande variété de trajectoires parcourues dans les cas où N est grand.

On peut également calculer la "root mean square error" de nos estimateurs en faisant varier le nombre de particules utilisé. Pour faire ce calcul, on fixe la racine de nos tirages aléatoires afin de rester cohérents.

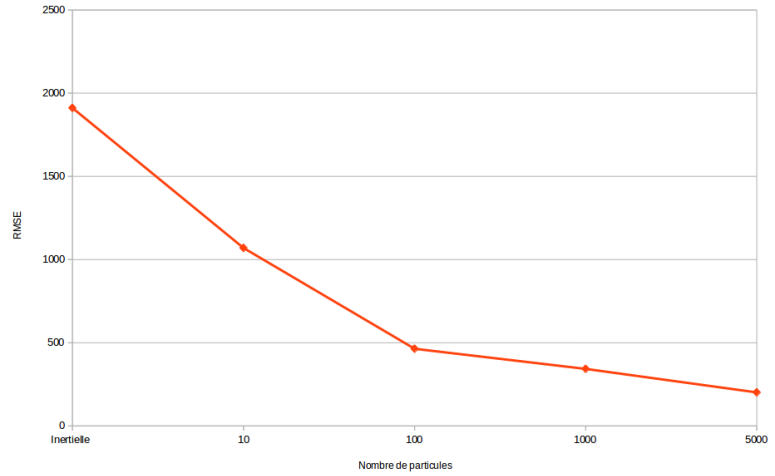


Figure 8: Evolution de la mean square error en fonction de l'évolution du nombre de particules

On voit que plus le nombre de particules augmente, plus l'erreur quadratique moyenne diminue.

2.3.2 Suivi du nuage pour un N fixé

On fixe N à 1000 et on trace le nuage de particules à chaque instant entre 1 et T . On remarque que ces particules restent concentrées autour de la trajectoire réelle.

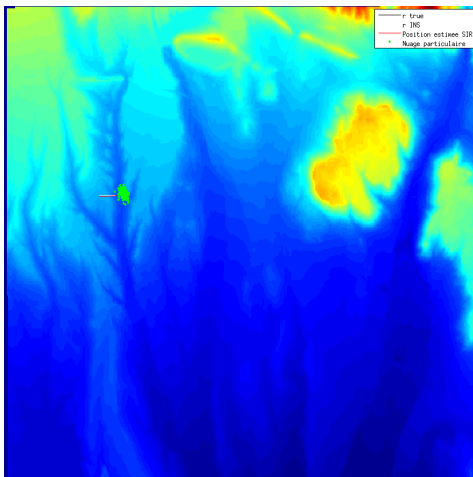


Figure 9: $t=10$

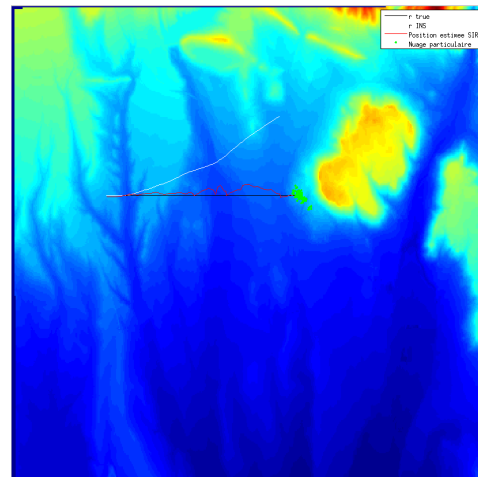


Figure 10: $t=70$

3 Algorithme hybride SIR/SIS

3.1 L'implémentation du SIR/SIS

Afin de montrer l'efficacité de l'algorithme SIR tel qu'il est décrit plus haut, on crée un algorithme qui utilise SIR selon une condition sur la taille effective de l'échantillon (permet de mesurer la qualité de l'échantillon). On implémente donc l'algorithme suivant. On note c la variable qui permet de calibrer la taille effective limite :

On **initialise** en tirant N particules initiales $(\xi_0^i)_{i \in \{1, \dots, N\}}$ suivant la loi

$$\xi_1 = \begin{pmatrix} \delta r_1 \\ \delta v_1 \end{pmatrix} \sim \mathcal{N}\left(0, \begin{pmatrix} \sigma_0 I_2 & 0 \\ 0 & \sigma \delta v_0 I_2 \end{pmatrix}\right) \quad (3.1)$$

```

for  $k \in \{2, \dots, T+2\}$  do
  On calcule :  $w_c = (\text{vraisemblance}((\xi_k^i), 1))_{i \in \{1, \dots, N\}}$ 
  On calcule :  $w_{sis} = w_{k-2} * w_c$ 
  On calcule :  $w_{sis} = \left(\frac{w_{sis}^i}{\sum_{i \in \{1, \dots, N\}} w_{sis}^i}\right)_{i \in \{1, \dots, N\}}$ 
  On calcule :  $N_{eff} = \frac{1}{\sum_{i \in \{1, \dots, N\}} w_{sis}^2}$ 
  if  $N_{eff} \leq c.N$  then
    On calcule :  $p_{k-1} = \left(\frac{w_c^i}{\sum_{i \in \{1, \dots, N\}} w_c^i}\right)_{i \in \{1, \dots, N\}}$ 
    On calcule :  $\hat{\xi}_{k-1} = \text{fct-multi}(\xi_{k-1}, w_{k-1}, N)$ 
    On calcule :  $\xi_k = (\text{update-delta}(\hat{\xi}_{k-1}^i))_{i \in \{1, \dots, N\}}$ 
    On stocke la valeur estimée :  $\bar{\xi}_{k-1} = p_{k-1}^T \xi_{k-1}$ 
    On stocke la valeur pour le calcul du SIS suivant  $w_{k-1} = \left(\frac{1}{N}\right)_{i \in \{1, \dots, N\}}$ 
  else
    On calcule :  $p_{k-1} = w_{sis}$ 
    On calcule :  $\xi_k = (\text{update-delta}(\xi_{k-1}^i))_{i \in \{1, \dots, N\}}$ 
    On stocke la valeur estimée :  $\bar{\xi}_{k-1} = p_{k-1}^T \xi_{k-1}$ 
    On stocke la valeur pour le calcul du SIS suivant  $w_{k-1} = w_{sis}$ 
end

```

Le vecteur décrivant la trajectoire est donné par :

$$(r_1^{INS} + \bar{\xi}_1(1:2), \dots, r_k^{INS} + \bar{\xi}_k(1:2), \dots, r_{T+1}^{INS} + \bar{\xi}_{T+1}(1:2)) \quad (3.2)$$

Algorithm 2: Algorithme hybride SIR SIS dépendant de la taille effective

3.2 Résultats SIS seul

En utilisant l'algorithme SIS ($c=0$) on remarque que les particules divergent parce qu'elles ne sont pas redistribuées.

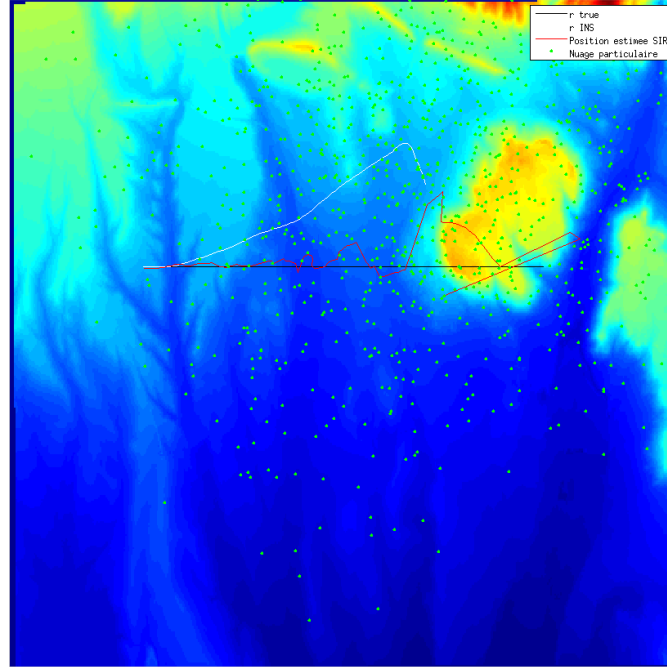


Figure 11: l'état final des particules - $N=1000$

Le résultat obtenu en utilisant SIS dégrade donc le résultat que l'on obtenait en utilisant SIR.

3.3 Résultats hybride SIS/SIR

Les résultats varient avec le paramètre c qui permet de sélectionner selon la taille effective entre l'algorithme SIR ou SIS. On voit que la distribution des particules est bien mieux concentrée lorsque c est proche de 1 et que l'on utilise l'algorithme SIR.

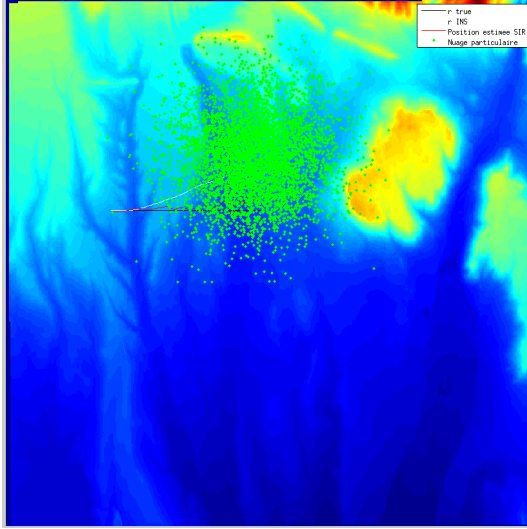


Figure 12: $c=0$, $N=5000$, $t=50$

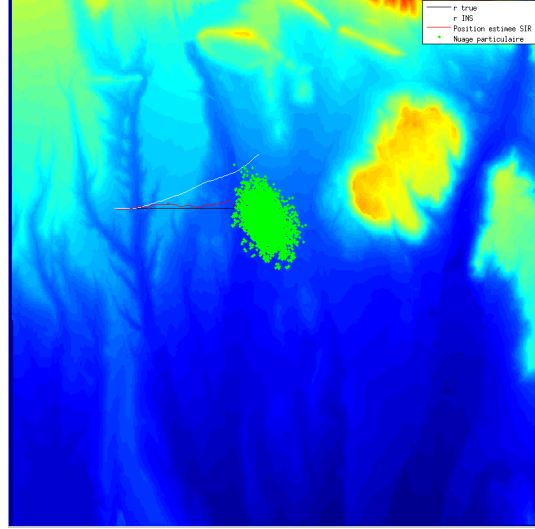


Figure 13: $c=0.2$, $N=5000$, $t=50$

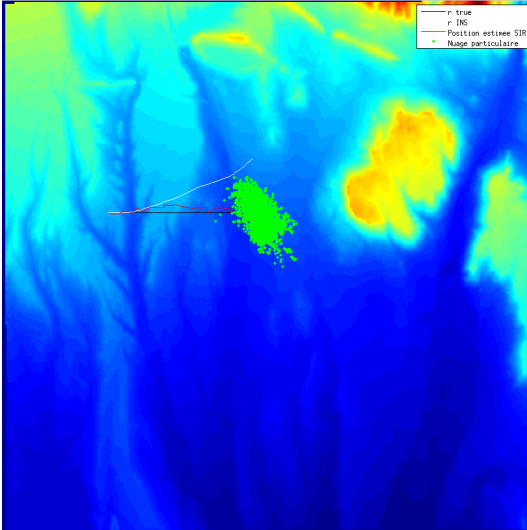


Figure 14: $c=0.5$, $N=5000$, $t=50$

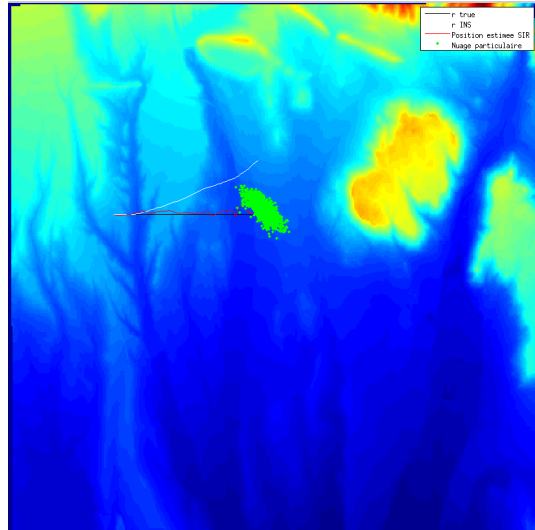


Figure 15: $c=1$, $N=5000$, $t=50$

On voit que pour c faible (de l'ordre de 0) les particules ont tendances à aller n'importe où, il arrive même que les particules sortent de la carte ce qui provoque des erreurs (on gère cette erreur en faisant en sorte que la vraisemblance de la particule soit nulle si on est en dehors de la carte). On voit que la redistribution a toute son utilité dans notre estimation.

On peut tracer l'évolution de la RMSE en fonction de l'évolution du c pour voir l'efficacité de l'algorithme lorsque le paramètre de choix entre SIR/SIS augmente (plus c est proche de 1 plus on utilise SIR).

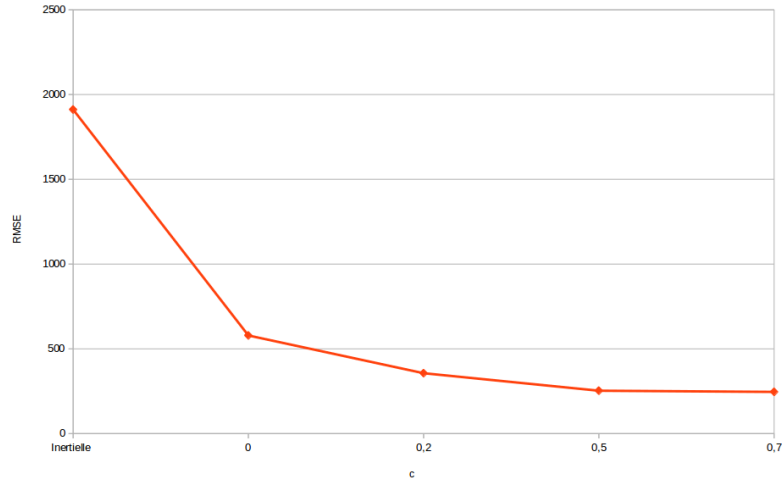


Figure 16: Evolution de la mean square error en fonction du paramètre c pour $N = 1000$ fixé

Cette courbe nous permet de voir que plus on utilise SIS moins les performances de l'algorithme sont bonnes. On voit tout de même que l'utilisation de SIS seul permet déjà d'améliorer le résultat basé seulement sur la mesure inertielle.

4 Conclusion

A travers ce premier TP du cours de Filtrage Bayésien, nous avons pu implémenter et ainsi mieux comprendre les concepts vus en cours. Nous avons traité un cas pratique de correction d'une trajectoire estimée à partir d'une mesure inertielle grâce à l'utilisation de la mesure de l'altitude de l'avion et d'une carte contenant les reliefs connus du paysage parcouru.

On a ainsi pu voir que l'algorithme avec redistribution multinomiale SIR permet largement d'améliorer les résultats. L'erreur quadratique moyenne assez faible ainsi que la comparaison avec l'algorithme nous permet de conclure quant à la pertinence de l'utilisé d'algorithmes basés sur SIR.