

# SOD333 - Filtrage bayésien et approximation particulière : Recalage altimétrique de navigation inertielle

Farès EL BOUZIANI - Ilyes ER-RAMMACH

19 octobre 2020

## 1 Présentation du problème

Un avion survole une zone dont le relief est connu : la hauteur  $h(r)$  du relief en chaque point de coordonnée horizontale est connu, et enregistrée dans une carte numérique fournie dans le fichier *mnt.mat*. Dans la suite, la position horizontale de l'avion est notée  $r$ , la position verticale est notée  $z$ , et la vitesse horizontale est notée  $v$ . A l'instant 0, la position horizontale initiale de l'avion est  $r_0$ , son altitude initiale est  $z_0$  et sa vitesse horizontale initiale est  $v_0$ . En réalité, l'avion se déplace à l'altitude  $z = z_0$  constante et à la vitesse horizontale  $v = v_0$ . Un exemple de trajectoire est fournie dans le fichier *traj.mat*. Le but est alors d'estimer efficacement la trajectoire réelle de l'avion. En effet on verra que la mesure inertielle ne correspond pas du tout à la trajectoire réelle. Comment peut-on corriger cette mesure pour avoir un résultat plus satisfaisant ?

## 2 Trajectoire réelle

Voici en rouge la trajectoire réelle de l'avion, notée  $r_k^{\text{true}}$ , ou plus simplement  $r_k$ , a priori inconnue :

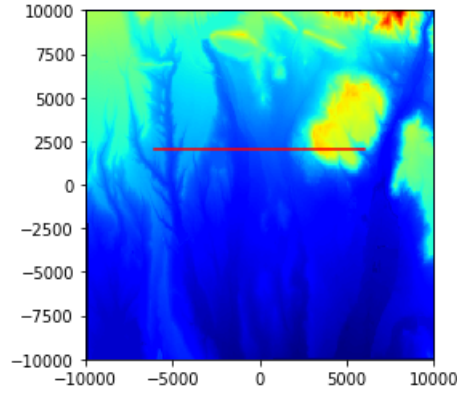


FIGURE 1 – Tracé de la trajectoire réelle (depuis les données fournies)

## 3 Navigation : estimation inertielle

Voici en rose le tracé de la mesure inertielle, notée  $r_k^{\text{INS}}$  faite sur la trajectoire (à l'aide des gyroscopes de l'avion). On voit que les deux mesures ne correspondent

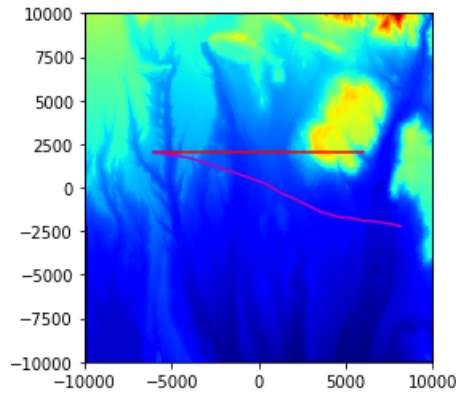


FIGURE 2 – Tracés de la trajectoire inertielle et la trajectoire réelle

pas du tout. On introduit alors les variables  $\delta r_k = r_k - r_k^{\text{INS}}$  et  $\delta v_k = v_k - v_k^{\text{INS}}$ .

## 4 Equation d'état

En reprenant les équations  $r_k^{\text{INS}} = r_{k-1}^{\text{INS}} + \Delta v_{k-1}^{\text{INS}}$ ,  $v_k^{\text{INS}} = v_{k-1}^{\text{INS}} + \Delta a_{k-1}^{\text{INS}}$ , et enfin  $a_k^{\text{INS}} = a_k + w_k^{\text{INS}}$ , on a en soustrayant :

$$\delta r_k = r_{k-1} + \Delta v_{k-1} - (r_{k-1}^{\text{INS}} + \Delta v_{k-1}^{\text{INS}})$$

$$\delta v_k = v_{k-1} + \Delta a_{k-1} - (v_{k-1}^{\text{INS}} + \Delta a_{k-1}^{\text{INS}})$$

Pour la première équation, il suffit de reprendre l'écriture "différentielle" et on obtient :

$$\delta r_k = \delta r_{k-1} + \Delta \delta v_{k-1}$$

Pour la deuxième, il suffit d'utiliser l'équation de l'accélération et il vient

$$\delta v_k = \delta v_{k-1} - \Delta w_k^{\text{INS}}$$

Autrement dit, en écriture matricielle :

$$\begin{pmatrix} \delta r_k \\ \delta v_k \end{pmatrix} = \begin{pmatrix} \delta r_{k-1} + \Delta \delta v_{k-1} \\ \delta v_{k-1} \end{pmatrix} - \Delta \begin{pmatrix} 0 \\ w_k^{\text{INS}} \end{pmatrix}$$

i.e.

$$\begin{pmatrix} \delta r_k \\ \delta v_k \end{pmatrix} = \begin{pmatrix} I_2 & \Delta I_2 \\ 0 & I_2 \end{pmatrix} \begin{pmatrix} \delta r_{k-1} \\ \delta v_{k-1} \end{pmatrix} - \Delta \begin{pmatrix} 0 \\ w_k^{\text{INS}} \end{pmatrix}$$

qui sera notre équation d'état.

## 5 Hauteur du relief

On représente sur un même graphique le profil réel du terrain survolé  $h(r_k)$  (courbe bleu) ainsi que les mesures bruitées de la hauteur du relief survolé  $h_k^{\text{ALT}}$  (points rouges) recueillis comme indiqué dans l'énoncé : On remarque que les deux courbes sont très proches, les mesures bruitées sont donc assez précises et seront efficaces pour corriger les mesures bruitées de la position horizontale par les méthodes vues dans le cours (SIR...). Pour obtenir le profil exact de la hauteur du relief, sous Python, on a créé une fonction  $h(x, y)$  qui permet de récupérer la hauteur du relief au point de coordonnées horizontales  $(x, y)$ . En effet, la carte étant en 2D, l'enjeu ici était d'obtenir  $i$  et  $j$  (coordonnées sur la carte) en fonction de  $x$  par un jeu de changement d'échelle (sachant que  $i$  va de 0 à  $N_2 - 1$  et  $j$  de 0 à  $N_1 - 1$ ), en prenant la partie entière et enfin en prenant garde au fait qu'on lise une matrice de haut

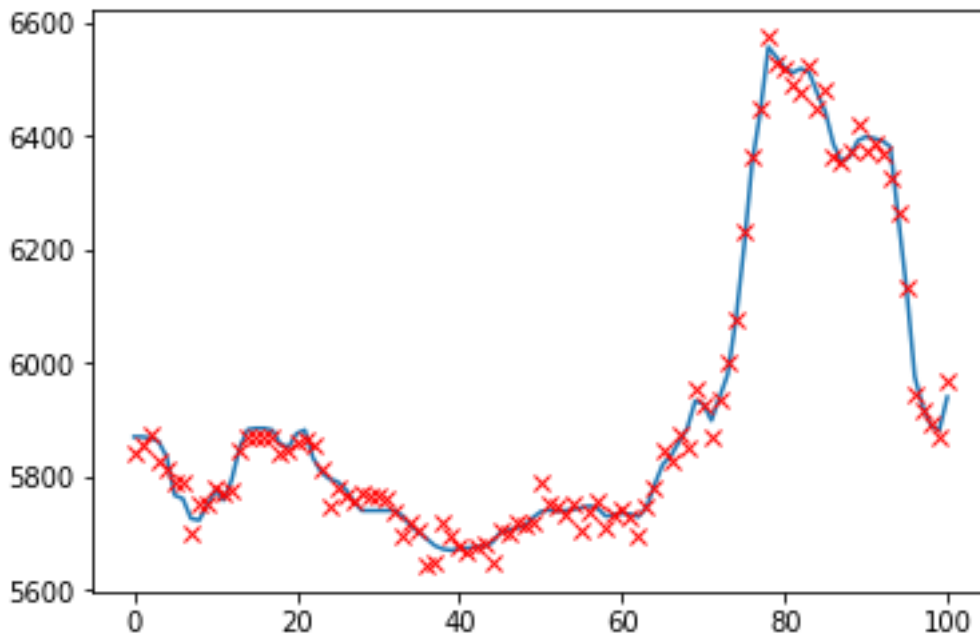


FIGURE 3 – Profil réel du terrain survolé et mesures altimétriques

en bas et non de bas en haut (on "inverse" alors la coordonnée verticale lors du changement d'échelle), ce qui donne :

$$i = \left\lfloor N_2 \frac{X_{2,\max} - y}{X_{2,\max} - X_{2,\min}} \right\rfloor$$

$$j = \left\lfloor N_1 \frac{x - X_{1,\min}}{X_{1,\max} - X_{1,\min}} \right\rfloor$$

Maintenant qu'on sait retrouver la hauteur d'un point quelconque du plan, il suffit de créer le vecteur  $h_k = h(r_k^{\text{true}})$  pour obtenir le profil exact du relief survolé par l'avion.

## 6 Equation d'observation

On a d'une part  $z^{\text{BAR}} = z_k + w_k^{\text{BAR}}$  et d'autre part  $d_k^{\text{ALT}} = z_k - h(r_k) + w_k^{\text{ALT}}$ . En partant de l'expression basique de  $h_k^{\text{ALT}}$ , on a :

$$h_k^{\text{ALT}} = z_k^{\text{BAR}} - d_k^{\text{ALT}}$$

On utilise la première équation pour substituer  $z_k^{\text{BAR}}$  et on trouve :

$$h_k^{\text{ALT}} = z_k - d_k^{\text{ALT}} + w_k^{\text{BAR}}$$

Ensuite, on utilise la deuxième équation pour substituer  $d_k^{\text{ALT}}$  et il vient :

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

$$h_k^{\text{ALT}} = h(r_k) + w_k^{\text{BAR}} - w_k^{\text{ALT}}$$

et enfin en utilisant  $r_k = r_k^{\text{INS}} + \delta r_k$  on a finalement l'équation d'observation recherchée :

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

La fonction de vraisemblance est la densité du bruit d'observation, c'est-à-dire la densité de  $w_k^{\text{BAR}} - w_k^{\text{ALT}}$  qui n'est autre que la densité d'une gaussienne centrée de variance  $\sigma_{\text{BAR}}^2 + \sigma_{\text{ALT}}^2$  (par somme de gaussienne indépendante). Elle est en particulier indépendante de  $k$ .

## 7 Filtrages

Nous avons programmé trois algorithmes : le filtrage SIR, le filtrage SIS, et le filtrage adaptatif. Le dernier est en fait un cas plus général des deux précédents, mais le lecteur est tout de même invité à tous les tester.

### 7.1 Discussion : filtrage SIR

La programmation de cet algorithme n'a pas posé de problème particulier et permet une correction intéressante de la mesure inertielle de la position. Pour le programmer, il nous a suffi de suivre les grandes lignes de l'algorithme général présenté dans le cours. Voici des résultats pour différents nombres  $N$  de particules : Il semblerait que la précision est plus fine pour des grandes valeurs de  $N$ .

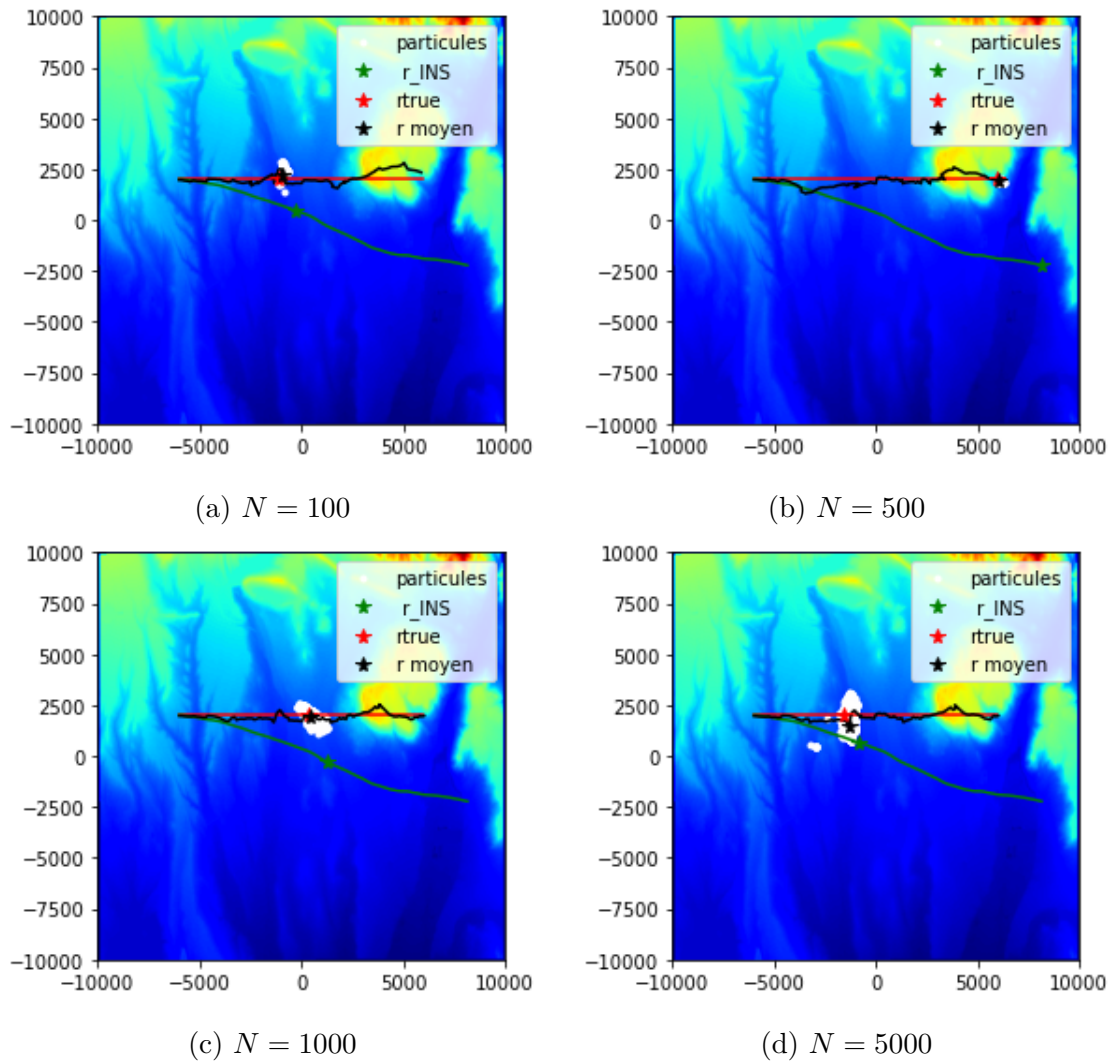


FIGURE 4 – Filtrage SIR

## 7.2 Discussion : filtrage SIS

La programmation de ce filtrage a été un peu plus problématique. En effet, sans rééchantillonnage, les particules sont libres de se propager sans contrainte : elles peuvent en particulier sortir de la carte et entraîner une erreur, ce qui arrive en fait à coup sûr dans notre cas. Pour contourner le problème, nous sommes revenus à la définition de la fonction  $h(x, y)$ . Nous calculons les indices  $i$  et  $j$  comme indiqué dans le paragraphe "Hauteur du relief", puis nous projetons les indices sur leur intervalle de définition respectif. Par exemple, si  $j < 0$ , alors on pose  $j = 0$ . Si  $j > N_1 - 1$ , alors on pose  $j = N_1 - 1$ , et de même pour  $i$ . Ceci entraîne éventuellement une mesure erronée de la hauteur du relief au point du plan horizontal  $(x, y)$ , mais étant donné que les poids associés à ces points sont généralement très petits, l'erreur ne sera pas gênante. Cette manipulation nous a permis de faire fonctionner l'algorithme SIS, mais le résultat obtenu est nettement moins satisfaisant que celui de l'algorithme SIR. En effet, les particules sont bien plus dispersées et le chemin corrigé est moins proche du chemin réel. Voici les résultats obtenus :

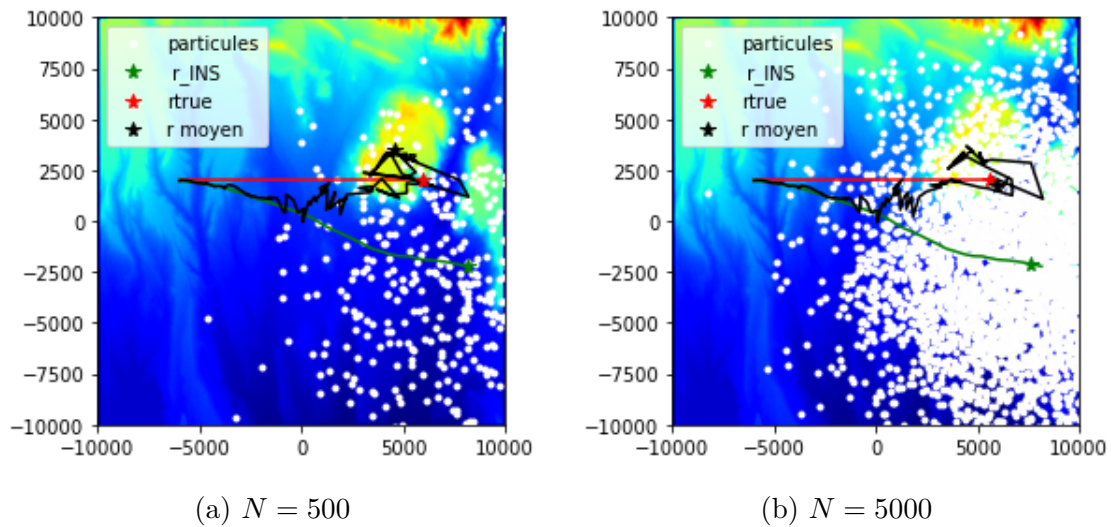


FIGURE 5 – Filtrage SIS

## 7.3 Discussion : filtrage adaptatif

Ce dernier étant très proche des deux précédents, nous avons peu à dire sur sa façon de le programmer. Les résultats obtenus sont d'autant plus précis que  $c$  est proche de 1, ce qui est finalement logique dans notre cas. Voici nos résultats en faisant varier  $c$ , et en gardant un nombre de particules égale à  $N = 500$ .

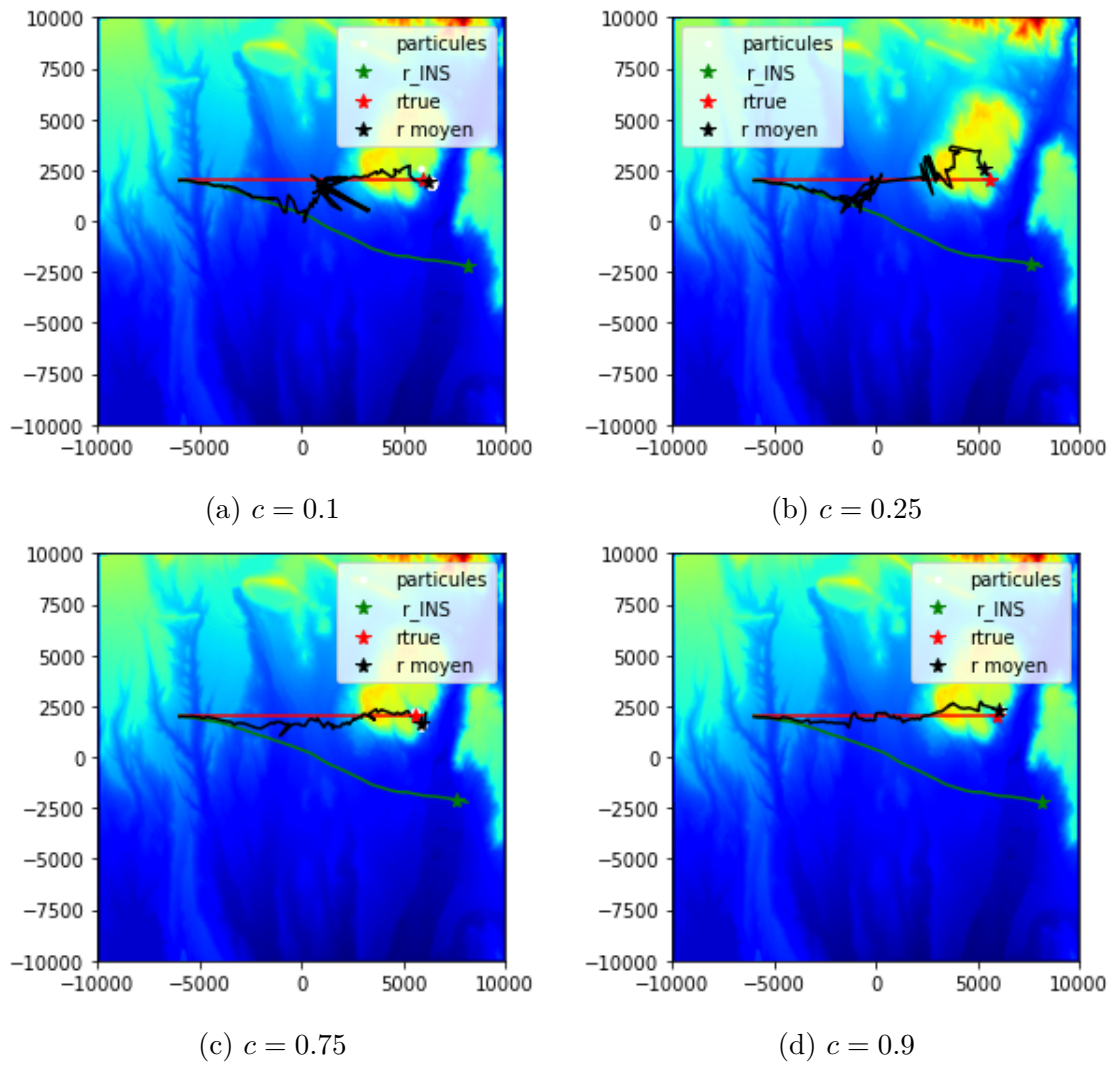


FIGURE 6 – Filtrage adaptatif

## 8 Conclusion

Le meilleur algorithme semble le celui du filtrage SIR, ou encore celui du filtrage adaptatif avec un paramètre  $c$  assez proche de 1. Le filtrage SIR et adaptatif sont donc de très bons programmes de filtrage pour corriger des mesures bruitées de position par exemple, ou encore de vitesse. Nous invitons le lecteur à tester les codes afin d'accéder notamment aux animations montrant les trajectoires de l'avion calculées selon les algorithmes utilisés.