

ENSTA — Filtrage bayésien et approximation particulière

TP: Recalage altimétrique de navigation inertielle

vendredi 2 et vendredi 9 octobre 2020

Un avion survole une zone dont le relief est connu : la hauteur $h(r)$ du relief en chaque point de coordonnée horizontale r est connue, et enregistrée dans une carte numérique fournie dans le fichier `mnt.mat`.

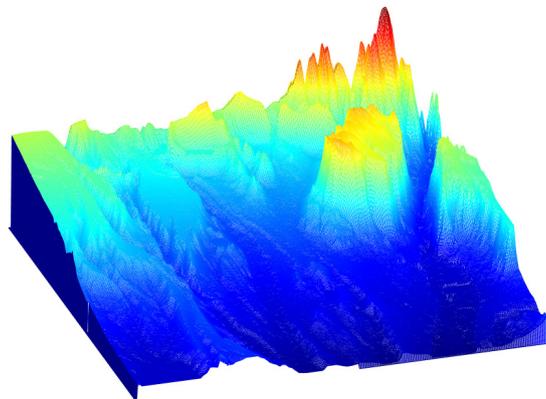


Figure 1: Relief 3D de la zone survolée

Pour récupérer ces informations

```
map = io.loadmat('mnt.mat')['map']  
N1 = map.shape[1]  
N2 = map.shape[0]
```

et pour la visualisation

```
plt.imshow(map, cmap='jet', extent=[X1MIN, X1MAX, X2MIN, X2MAX])
```

La liste des modules Python importés et les valeurs numériques des constantes sont données à la fin de l'énoncé.

Dans la suite, la position horizontale de l'avion est notée r , la position verticale, ou altitude, 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 constante $v = v_0$. Un exemple de trajectoire est fournie dans le fichier `traj.mat`. Pour récupérer ces informations

```

traj = io.loadmat('traj.mat')
rtrue = traj['rtrue']
vtrue = traj['vtrue']

```

et pour la visualisation

```

plt.plot(rtrue[0,:],rtrue[1,:], 'r-')

```

- (i) Représenter pour chaque instant entre 0 et T , la position horizontale exacte r_k de l'avion.

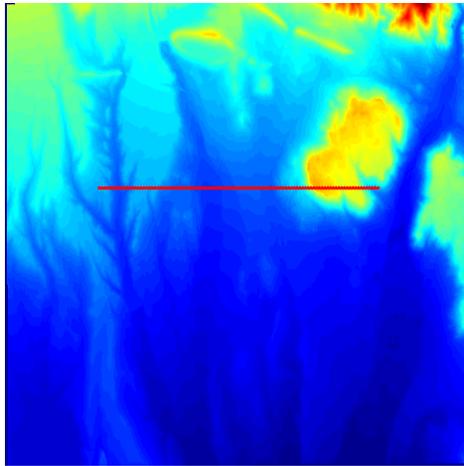


Figure 2: Modèle numérique de terrain, et trajectoire réelle

Pour effectuer la navigation, c'est-à-dire pour permettre à l'avion d'estimer sa propre position horizontale r_k et sa propre vitesse horizontale v_k à chaque instant t_k , on recueille (au moyen d'accéléromètres et de gyroscopes installés à bord) avec un pas de temps $\Delta = t_k - t_{k-1}$ et jusqu'à l'instant final T , l'accélération horizontale de l'avion avec une erreur additive modélisée par un bruit blanc gaussien centré de matrice de covariance $\sigma_{\text{INS}}^2 I_2$ (on dénote par I_2 la matrice identité de dimension 2×2). L'écart-type σ_{INS} est une caractéristique supposée connue de la centrale de navigation inertielle utilisée. En d'autres termes

$$a_k^{\text{INS}} = a_k + w_k^{\text{INS}} ,$$

où a_k dénote l'accélération horizontale réelle de l'avion, et où la suite w_k^{INS} est un bruit blanc gaussien centré de matrice de covariance $\sigma_{\text{INS}}^2 I_2$.

Les mesures inertielles bruitées $(a_1^{\text{INS}}, \dots, a_{n_{\text{max}}-1}^{\text{INS}})$ sont recueillies par l'avion le long de sa trajectoire, et sont fournies dans le fichier `ins.mat`. À partir de ces mesures inertielles, et à partir de la position initiale r_0 et de la vitesse initiale v_0 , on obtient l'estimation r_k^{INS} de la position

horizontale obtenue en intégrant les mesures d'accélération horizontale à l'aide du modèle d'état suivant

$$\begin{pmatrix} r_k^{\text{INS}} \\ v_k^{\text{INS}} \end{pmatrix} = \begin{pmatrix} I_2 & \Delta I_2 \\ 0 & I_2 \end{pmatrix} \begin{pmatrix} r_{k-1}^{\text{INS}} \\ v_{k-1}^{\text{INS}} \end{pmatrix} + \Delta \begin{pmatrix} 0 \\ a_k^{\text{INS}} \end{pmatrix} \quad \text{avec} \quad \begin{pmatrix} r_1^{\text{INS}} \\ v_1^{\text{INS}} \end{pmatrix} = \begin{pmatrix} r_0 \\ v_0 \end{pmatrix} .$$

Pour récupérer ces informations

```
a_INS = io.loadmat('ins.mat')['a_INS']
```

pour le calcul de la trajectoire estimée à partir des mesures INS seulement

```
r_INS = np.zeros(rtrue.shape)
v_INS = np.zeros(vtrue.shape)

r_INS[:,0] = r0
v_INS[:,0] = v0
for k in range(1,nmax):
    r_INS[:,k] = r_INS[:,k-1]+delta*v_INS[:,k-1]
    v_INS[:,k] = v_INS[:,k-1]+delta*a_INS[:,k-1]
```

et pour la visualisation

```
plt.plot(r_INS[0,:],r_INS[1,:], 'm-')
```

- (ii) Représenter sur le même graphique pour chaque instant entre 0 et T , la position horizontale exacte r_k de l'avion, et son estimation inertielle r_k^{INS} . Que remarque-t-on ?

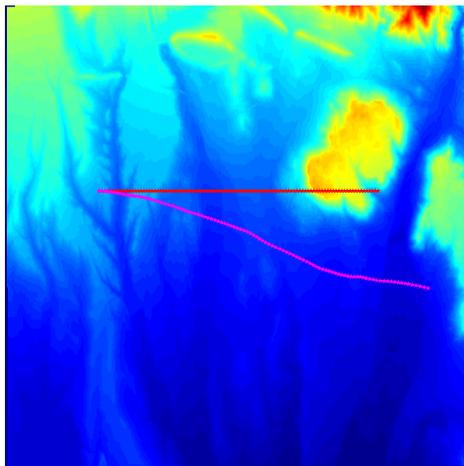


Figure 3: Modèle numérique de terrain, trajectoire réelle et trajectoire inertielle

- (iii) On introduit comme nouvelles variables d'état les erreurs d'estimation inertielle $\delta r_k = r_k - r_k^{\text{INS}}$ sur la position horizontale et $\delta v_k = v_k - v_k^{\text{INS}}$ sur la vitesse horizontale. Montrer que le modèle d'état correspondant est donné par

$$\begin{pmatrix} \delta r_k \\ \delta v_k \end{pmatrix} = \begin{pmatrix} I_2 & \Delta I_2 \\ 0 & I_2 \end{pmatrix} \begin{pmatrix} \delta r_{k-1} \\ \delta v_{k-1} \end{pmatrix} - \Delta \begin{pmatrix} 0 \\ w_k^{\text{INS}} \end{pmatrix} \quad \text{avec} \quad \begin{pmatrix} \delta r_1 \\ \delta v_1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix},$$

où la suite w_k^{INS} est un bruit blanc gaussien centré de matrice de covariance $\sigma_{\text{INS}}^2 I_2$.

Pour corriger la dérive de l'estimation inertielle r_k^{INS} de la position horizontale par rapport à la position horizontale exacte r_k , on recueille séparément (au moyen d'un radar altimétrique, ou radio-altimètre, installé à bord) avec le même pas de temps Δ une mesure d_k^{ALT} de la hauteur de l'avion au-dessus du relief situé à la verticale, avec une erreur additive modélisée par un bruit blanc gaussien centré de variance σ_{ALT}^2 . L'écart-type σ_{ALT} est une caractéristique supposée connue du radio-altimètre utilisé.

On recueille également (au moyen d'un baromètre altimétrique, ou baro-altimètre, installé à bord) avec le même pas de temps Δ une mesure z_k^{BAR} de l'altitude de l'avion, avec une erreur additive modélisée par un bruit blanc gaussien centré de variance σ_{BAR}^2 . L'écart-type σ_{BAR} est une caractéristique supposée connue du baro-altimètre utilisé.

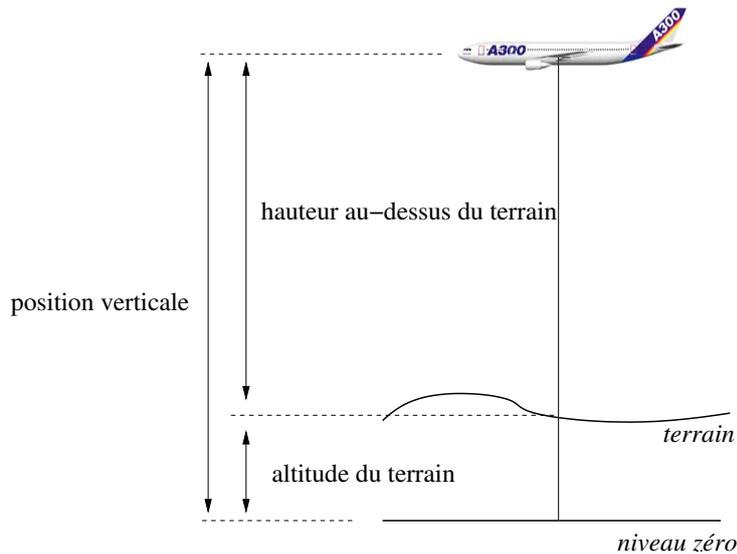


Figure 4: Principe du recalage altimétrique

- (iv) A chaque instant t_k , lire sur la carte la hauteur $h(r_k)$ du relief au point de coordonnée horizontale r_k situé à la verticale de l'avion. Représenter graphiquement dans un plan vertical le profil du relief survolé.

A chaque instant t_k , le radio–altimètre fournit une mesure bruitée d_k^{ALT} de la distance entre l’avion et le relief, c’est-à-dire

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

où r_k dénote la position horizontale réelle de l’avion, où z_k dénote l’altitude réelle de l’avion, et où la suite w_k^{ALT} est un bruit blanc gaussien centré de variance σ_{ALT}^2 . Au même instant t_k , le baro–altimètre fournit une mesure bruitée z_k^{BAR} de l’altitude de l’avion, c’est-à-dire

$$z_k^{\text{BAR}} = z_k + w_k^{\text{BAR}} ,$$

où z_k dénote l’altitude réelle de l’avion, et où la suite w_k^{BAR} est un bruit blanc gaussien centré de variance σ_{BAR}^2 .

La hauteur du relief survolé à l’instant t_k déduite à partir des mesures fournies par le radio–altimètre et par le baro–altimètre est donc

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

Les mesures radio–altimétriques bruitées ($d_1^{\text{ALT}}, \dots, d_{n_{\text{max}}}^{\text{ALT}}$) et les mesures baro–altimétriques bruitées ($z_1^{\text{BAR}}, \dots, z_{n_{\text{max}}}^{\text{BAR}}$) sont recueillies par l’avion le long de sa trajectoire, les mesures bruitées ($h_1^{\text{ALT}}, \dots, h_{n_{\text{max}}}^{\text{ALT}}$) de la hauteur du relief survolé sont ensuite calculées à bord de l’avion, et sont fournies dans le fichier `alt.mat`. Pour récupérer ces informations

```
h_ALT = io.loadmat('alt.mat')['h_ALT'][0]
```

- (v) Représenter sur le même graphique pour chaque instant entre 0 et T le profil exact du relief survolé et les mesures bruitées de la hauteur du relief survolé.

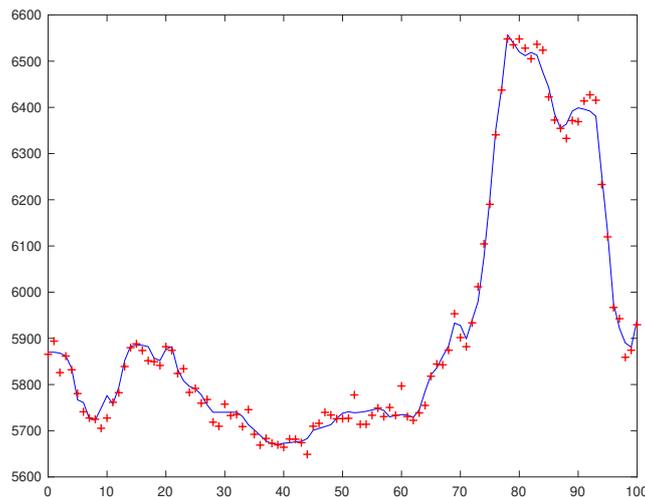


Figure 5: Profil réel du terrain survolé et mesures altimétriques

- (vi) Montrer que la mesure bruitée h_k^{ALT} fournie par le radio–altimètre et par le baro–altimètre peut être reliée à l’erreur d’estimation inertielle δr_k sur la position horizontale par

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

En déduire l’expression de la fonction de vraisemblance.

En résumé, le modèle d’état utilisé pour le recalage altimétrique de navigation inertielle comprend :

- l’équation d’état

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

où la suite w_k^{INS} est un bruit blanc gaussien centré de variance $\sigma_{\text{INS}}^2 I_2$,

- la condition initiale incertaine

$$\begin{pmatrix} \delta r_0 \\ \delta v_0 \end{pmatrix} \text{ gaussienne, centrée, de matrice de covariance } \begin{pmatrix} \sigma_{r_0}^2 I_2 & 0 \\ 0 & \sigma_{v_0}^2 I_2 \end{pmatrix} ,$$

- et l’équation d’observation

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

où la suite w_k^{ALT} est un bruit blanc gaussien centré de variance σ_{ALT}^2 , et où la suite w_k^{BAR} est un bruit blanc gaussien centré de variance σ_{BAR}^2 .

L’estimation inertielle r_k^{INS} de la position horizontale fournie par la centrale inertielle, et la mesure h_k^{ALT} de la hauteur du relief fournie par le radio–altimètre et par le baro–altimètre sont disponibles. La fonction $r \mapsto h(r)$ n’est pas connue de façon analytique, mais définie *point–par–point* en allant lire la carte numérique.

- (vii) Mettre en œuvre un filtre particulaire SIR (avec redistribution multinomiale à chaque pas de temps) pour l’estimation récursive à chaque instant entre 0 et T de l’état caché $(\delta r_k, \delta v_k)$ à partir des mesures $(h_0^{\text{ALT}}, \dots, h_k^{\text{ALT}})$. En gardant les mêmes observations pour faciliter la comparaison, répéter l’expérience en faisant varier le nombre N de particules : essayer d’abord $N = 1000$, puis essayer $N = 5000$, et enfin $N = 500$ et $N = 100$.

A chaque instant entre 0 et T , représenter sur un même graphique, la position horizontale r_k de l’avion, l’estimation inertielle r_k^{INS} de la position horizontale et le nuage de particules.

On pourra aussi représenter la moyenne empirique pondérée des positions des particules, et réaliser une animation pour visualiser l’évolution du nuage de particules au cours du temps.

Pour l’étape de redistribution, on pourra utiliser la fonction Python `resampling_multi` fournie dans le fichier `resampling_multi.py`. La syntaxe est

```
indices = resampling_multi(weight,N)
```

Cette fonction fournit les indices, à valeurs dans $\{1, \dots, N\}$, des N particules qui sont sélectionnées en fonction de leurs poids respectifs, rangés dans le tableau `weight`.

- (viii) Pour mettre en évidence l'effet de la redistribution, mettre en œuvre un filtre particulaire SIS (sans redistribution), ou encore mieux un filtre particulaire avec redistribution adaptative, où la redistribution n'est effectuée que si la taille effective N_{eff} de l'échantillon descend au-dessous du seuil cN , avec $0 \leq c \leq N$: le cas $c = 0$ correspond au filtre SIS, le cas $c = 1$ correspond au filtre SIR. En gardant les mêmes observations pour faciliter la comparaison, répéter l'expérience en faisant varier le seuil c de redistribution.

Les modules nécessaires sont

```
import numpy as np # calcul numerique
import numpy.random as rnd # fonctions pseudo-aleatoires
import matplotlib.pyplot as plt # fonctions graphiques a la MATLAB
import matplotlib.animation as anim # fonctions d'animation
import scipy.io as io # fonctions pour l'ouverture des fichiers .mat de MATLAB
import math
```

et les valeurs numériques des constantes sont données dans la Table 1 ci-dessous.

| | |
|----------------------------|--------------------|
| $[X_{1,\min}, X_{1,\max}]$ | $[-10000, +10000]$ |
| $[X_{2,\min}, X_{2,\max}]$ | $[-10000, +10000]$ |
| r_0 | $(-6000, +2000)$ |
| v_0 | $(+120, 0)$ |
| σ_{r_0} | 100 |
| σ_{v_0} | 10 |
| σ_{INS} | 7 |
| σ_{ALT} | 10 |
| σ_{BAR} | 20 |
| Δ | 1 |
| T | 100 |

Table 1: Valeurs numériques