Positionnement GPS à partir du signal RF

Yann Méneroux

11 décembre 2023 - 12 janvier 2024

L'objectif de ces travaux pratiques consiste à estimer avec Matlab les mesures brutes de pseudo-distances [1] à partir du signal radio-fréquence reçu par une radio logicielle. Un objectif secondaire visera à en déduire la position géographique de l'utilisateur à l'instant de la mesure. On dispose des données suivantes :

- ➔ **`SDRSharp_20221105_202954Z_1575420000Hz_IQ.wav`** : le fichier du signal radio-fréquence en format binaire, acquis avec un taux d'échantillonnage de 2.048 Mega-échantillons par seconde, le **`11/05/2022`** à **`20:30:11.738`** (heure du récepteur, synchronisée avec le système GPS à 1 ms près) sur l'esplanade sud du Château de Vincennes.
- ➔ **`igu22346_12.sp3`** : les éphémérides de la constellation GPS contenant les positions des satellites, et leurs erreurs d'horloges respectives, à l'instant de l'observation.
- ➔ **`lfsr.m`** : un script Matlab permettant d'émuler l'électronique interne du GPS.

Les coordonnées de référence du point relevé sont données ci-dessous (vérité terrain) en coordonnées cartésiennes ECEF (Earth-Centered, Earth Fixed) :

`X = 4202099.937 m, Y = 179059.028 m, Z = 4778941.838 m`

Travail à rendre pour le **19/01/2024** :

Envoyer à l'adresse email `yann.meneroux@ign.fr` avec l'objet [M2SAM] TP GPS, un unique fichier de script Matlab `<nom_binome_1-nom_binome_2>.m` contenant les réponses aux questions ci-après. Les réponses aux questions théoriques (ne nécessitant pas de code informatique) seront données sous forme de commentaires (%).

---

1. On désigne par *pseudo-distance* la mesure brute effectuée par le récepteur, afin de garder en tête que cette dernière doit être corrigée en prenant en compte un certain nombre de facteurs d'erreur, pour en faire une distance géométrique entre le récepteur et les satellites.

Les observations utilisées pour ce TP ont été collectées avec un PC portable classique, équipé du programme *SDR Sharp* connectée à une antenne patch active mono-fréquence ANN-MS via une clé de radio logicielle (*Software Defined Radio* ou SDR) NooElec NESDR SMArTee. Les éphémérides peuvent quant à elles être récupérées a posteriori sur le serveur de l'*International GNSS Service*.

Le travail est divisé en quatre sections : dans un premier temps, nous cherchons à montrer que le signal satellite est en théorie imperceptible pour les récepteurs situés à la surface terrestre, ce qui nous conduira à étudier dans une deuxième partie, les mécanismes de codes pseudo-aléatoires mis en oeuvre pour contourner le problème. L'estimation des pseudo-distances séparant les satellites du récepteur, fera l'objet d'une troisième partie, tandis que la dernière section sera consacrée au calcul de la position de l'utilisateur.

I. Du signal... et du bruit

La constellation GPS est composée de 32 satellites orbitant à une altitude de 20 000 km. Chaque satellite émet, à l'aide d'une antenne de gain 13 dBi (gain relatif à une antenne isotrope idéale) un signal de puissance 25 W sur une porteuse de fréquence $L_1 = 1575.42$ MHz.

Q1. On rappelle qu'une puissance $P_{[mW]}$ exprimée en milliwatts se convertit en dBm à l'aide de la formule :

$$P_{[dBm]} = 10 \log_{10} \left(\frac{P_{[mW]}}{1_{[mW]}} \right)$$

Convertir la puissance émise par les satellites en dBm.

Q2. On donne ci-dessous la formule de propagation de Friis, exprimant la puissance reçue par une utilisateur P^r, en fonction de la puissance émise P^e ainsi que des gains G^e et G^r des antennes employées respectivement pour l'émission et la réception :

$$P^r_{[dBm]} = P^e_{[dBm]} + G^e_{[dBi]} + G^r_{[dBi]} + 20 \log_{10} \left(\frac{\lambda}{4\pi d} \right)$$

où λ représente la longueur d'onde du signal transmis, et d est la distance de propagation.

Calculer, en dBm puis en mW, la puissance reçue au niveau de l'antenne du récepteur GPS.

Q3. Comparer cette puissance à celle du bruit thermique engendré par le récepteur, dont la densité spectrale de puissance (exprimée en W/Hz) vaut $k_B T$, où T est la température ambiante (en Kelvin) et $k_B \approx 1.38 \times 10^{-23}\ J.K^{-1}$ est la constante de Boltzmann. On considérera que la bande passante du récepteur est de l'ordre de 2 MHz.

Combien vaut le ratio signal sur bruit ? Que peut-on en déduire quant à la capacité du récepteur à detecter le signal GPS ?

II. Le code à la rescousse

Afin de garantir la détection par le récepteur, la porteuse du signal GPS est modulée par un code binaire pseudo-aléatoire, parfaitement connu et unique pour chaque satellite. La réplication du code au niveau du récepteur et sa corrélation avec le signal reçu permet :

- de séparer les signaux reçus par les différents satellites
- de dater précisemment les instants d'arrivée (et donc *in fine* les temps de transmission) de ces mêmes signaux.

La génération du code est effectée matériellement à l'aide d'un registre à décalage à rétroaction linéaire (LFSR, ou *Linear Feedback Shift Register*) à 10 bits. Concrètement, à chaque pas de temps, un certain nombre de bits du registre sont récupérés en parallèle, puis leur somme modulo 2 (*ou exclusif*, noté $\oplus$) est calculée et la valeur obtenue est insérée en tête de registre, décalant ainsi toutes les valeurs d'un rang vers la droite.

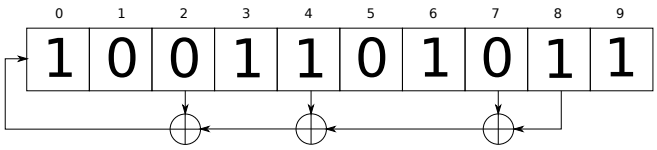

FIGURE 1 – Exemple de LFSR correspondant au polynôme de rétroaction $X^2 + X^4 + X^7 + X^8$.

Par exemple, sur la figure 1, à l'instant t, le registre contient la séquence $R_t = (1, 0, 0, 1, 1, 0, 1, 0, 1, 1)$. On calcule la somme (modulo 2) : $0 \oplus 1 \oplus 0 \oplus 1 = 0$, insérée en tête de registre, ce qui donne la séquence $R_{t+1} = (0, 1, 0, 0, 1, 1, 0, 1, 0, 1)$ au pas de temps suivant.

Q4. En considérant en particulier ce qu'il advient lorsque la séquence est initialisée avec les bits $R_0 = (0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$, montrer que le nombre maximal d'états distincts pouvant être atteint par le registre ne peut pas être supérieur à 1023.

Q5. On dit que la séquence générée est une *Maximal Length Sequence* (MLS), si elle atteint cette borne supérieure. C'est le cas des codes employés dans le système GPS, qui utilise deux LFSR de polynômes $X^2 + X^9$ et $X + X^2 + X^5 + X^7 + X^8 + X^9$, initialisés avec une séquence de 1, et générant respectivement les signaux G_1 et G_2, comme illustré sur la figure 2. Chaque satellite particularise alors ce signal, en récupérant deux bits spécifiques dans G_2, et en les sommant (modulo 2) pour fournir un signal S_2. Le signal pseudo-aléatoire final est alors obtenu par $G_1 \oplus S_2$.

Pour synthétiser ces codes en Matlab, nous utiliserons le script `lfsr.m`, contenant la fonction :

```
lfsr(init, feedback, N)
```

retournant `N` états d'un LFSR dont l'état initial est donné par le vecteur `init`, et dont les bits utilisés par la rétroaction sont indiqués dans un vecteur binaire `feedback`.

Par exemple, pour simuler les 5 pas de temps à venir du LFSR de l'image 1, on pourra écrire :

```
M = lfsr([1,0,0,1,1,0,1,0,1,1], [0,0,1,0,1,0,0,1,1,0], 5);
```

Par ailleurs, notons que si `feedback` décrit un LFSR de 10 bits de longueur maximale, alors l'instruction **`lfsr(init, feedback, 1023)(:,10)`**, *i.e.* la dernière colonne de la matrice calculée, permet d'obtenir le code pseudo-aléatoire généré en sortie du registre.

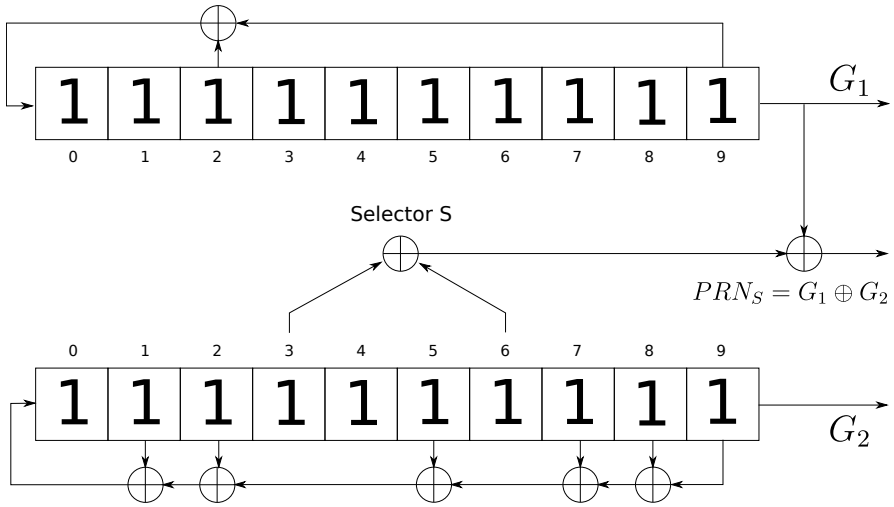

FIGURE 2 – Génération du code pseudo-aléatoire PRN pour le satellite S à l'aide de 2 LFSR. Ici, le sélecteur particularise G_2 en sélectionnant le couple de bits situés en 4^e et 7^e position, pour former S_2. Ce couple est distinct pour chaque satellite.

En utilisant (plusieurs fois) la fonction **`lfsr`**, écrire (dans un nouveau script) une fonction `cacode` prenant en entrée un numéro de satellite (entre 1 et 32) et retournant les 1023 bits de son code pseudo-aléatoire (PRN). On utilisera la table `tap` du script **`lfsr.m`**, listant les codes à sélectionner afin de former S_2 pour chaque satellite.

Q6. Exécuter l'instruction `cacode(1)` et vérifier que les 30 premiers bits de la réponse obtenue sont :

```
1 1 0 0 1 0 0 0 0 0 1 1 1 0 0 1 0 1 0 0 1 0 0 1 1 1 1 0 0 1...
```

Q7. Dans le reste de ce TP, nous allons travailler avec un signal échantillonné à 2.048 MHz. Dans le système GPS, les 1023 bits du code sont modulés dans le signal sur une période de 1 milliseconde. Le code généré par la fonction `cacode` doit donc être sur-échantillonné pour être comparable à celui du signal reçu. Cette opération pourra être réalisée avec l'instruction :

```
PRN = repelem(CODE, 2048)(1023:1023:end);
```

où `CODE` désigne le vecteur de 1023 bits généré par `cacode`, tandis que `PRN` est le code final sur-échantillonné, de taille 2048.

III. Mesure des pseudo-distances

On commence par charger les données du signal brut radio-fréquence reçu par le récepteur. Le fichier .wav est composée d'une entête de 44 octets, suivie de la liste des octets du signal, alternant entre les composantes en phase (I) et en quadrature (Q). Nous allons charger les 10 000 premiers échantillons, ce qui correspond approximativement [2] aux 10 premières millisecondes de l'enregistrement.

```
path = 'SDRSharp_20221105_202954Z_1575420000Hz_IQ.wav';

file_id = fopen(path);          % Ouverture du fichier wav
fread(file_id, 44);             % Suppression de l'entête de 44 octets
A = fread(file_id, 2e4);        % Lecture de 20 000 octets
I = A(1:2:size(A)(1));          % Composante en phase (I)
Q = A(2:2:size(A)(1));          % Composante en quadrature (Q)
signal = (I-127)+j*(Q-127);     % Centrage et formation du signal complexe
fclose(file_id);                % Fermeture du fichier wav
```

Q8. A l'aide de l'application ci-dessous, repérer les satellites potentiellement visibles lors de l'acquisition.

Visibilité des satellites dans le ciel
https://qzss.go.jp/redirect/gnssview.html

Q9. Nous allons dans un premier temps nous intéresser au satellite le plus au zénith lors de l'acquisition : **G25**.

En utilisant le script `cacode`, générer le code de ce satellite dans un vecteur `prn` de taille 2048. Afin de pouvoir effectuer la corrélation avec le signal RF, on centrera le code autour de la valeur 0 :

```
code = (cacode(prn)-0.5);
```

Q10. On rappelle que dans un contexte non-relativiste, la fréquence reçue en un point est fonction de la vitesse radiale relative de l'émetteur, et s'exprime par :

$$f_{rec} = \left(1 - \frac{v}{c}\right) f_{em}$$

où v désigne la vitesse relative entre l'émetteur et le récepteur, et c est la vitesse de la lumière dans le vide, égale à 299 792 458 m/s. On appelle *décalage doppler* la différence entre $\Delta = f_{em} - f_{rec}$.

Sachant que la vitesse radiale des satellites par rapport au centre de la Terre est au maximum de $\pm$ 500 m/s, calculer la plage de fréquences (autour de la fréquence nominale d'émission) dans laquelle on peut s'attendre à trouver le signal.

Q11. Pour commencer, nous allons chercher le code reçu par le satellite G25 dans un signal de fréquence doppler $\Delta = -1340$ Hz. Pour ce faire, on génère un vecteur de pas de temps `time`, à la fréquence d'acquisition du

2. Le code de 1023 bits s'étend sur une période de 1 ms, ce qui donne une durée d'environ 0.978 μs par bit : les 10 000 premiers échantillons correspondent donc aux 9.78 premières millisecondes du signal enregistré

récepteur (2.048 MHz), puis on génère une exponentielle complexe à la pulsation doppler : $\omega = 2\pi\Delta$, modulée par le code $A(25)$ du satellite G25 :

$$r_{25}^{\omega}(t) = A(25).\exp(-j\omega t)$$

En langage Matlab :

```
time = (1:size(code)(1))/2048000;
doppler = -1340;
replica = code.*exp(-2*j*pi*doppler*time)';
```

L'objectif consiste alors à calculer la corrélation entre cette réplique r (`replica`) et le signal s effectivement reçu (`signal`). On rappelle que convolution et multiplication sont analogues entre les espaces temporel et fréquentiel. Autrement dit, la corrélation de deux signaux temporels, peut être évaluée par la multiplication terme-à-terme de leurs transformées de Fourier respectives. Ou encore, en notant $\mathcal{F}[.]$ l'opérateur transformée de Fourier, on a :

$$r * s = \mathcal{F}^{-1}\Big[\mathcal{F}(r) \times \mathcal{F}(s)^*\Big]$$

Cette observation prend toute son importance quand on sait que $\mathcal{F}$ peut être implémentée efficacement [3] à l'aide de la transformation de Fourier rapide (FFT) en un temps $\Theta(n \log n)$.

En Matlab, on utilisera la fonction **`fft`** (prenant en entrée le signal à transformer, et la longueur effectivement utilisée pour la corrélation) et sa fonction réciproque : **`ifft`**. Pour deux signaux **`f`** et **`g`**, la corrélation **`c`** s'écrit :

```
c = abs((ifft(fft(f,l).*conj(fft(g,l)))));
```

où **`l`** est un entier désignant la longueur de corrélation.

Calculer et représenter avec la fonction `plot` la corrélation des signaux `signal` et `replica` sur une longueur de 10 000 échantillons.

Pour chercher la présence effective du code de G25 dans le signal on pourra appeler les instructions :

```
[A dt] = max(c); dt = mod(dt, 2048);
```

où `A` représente la valeur maximale de corrélation pour tous les décalages possibles, et `dt` représente le décalage atteignant ce max. Le signal étant périodique, la valeur de décalage optimale peut être prise modulo la longueur du signal.

Vérifier que le décalage `dt` obtenu pour le satellite G25 est de 619 échantillons.

3. Considérant qu'il y a deux FFT directes, une FFT inverse et une multiplication terme-à-terme à appliquer, la partie droite de l'équation précédente s'évalue en un temps $3 \times \Theta(n \log n) + \Theta(n) = \Theta(n \log n)$ contre $\Theta(n^2)$ pour le calcul direct de la corrélation.

Sachant que le code (échantillonné en 2048 points) est de période 1 ms, cette valeur se convertit en unité de temps par :

$$dt = \frac{619}{2048} \times 1 \text{ ms} = 0.30225 \text{ ms}$$

Q12. Réitérer l'expérience de la question Q11 en modifiant la fréquence doppler (par exemple en fixant $\Delta = 500$ Hz) et/ou le numéro du satellite visé. Vérifier qu'aucune corrélation n'est trouvée dans le signal avec ces nouveaux paramètres.

Q13. L'expérimentation effectuée à la question précédente montre que pour chaque satellite n recherché dans le signal reçu s, on doit balayer l'ensemble des fréquences doppler de la plage définie à la question Q10 ; pour chaque fréquence $\Delta = \frac{\omega}{2\pi}$, on génère la réplique r_n^ω et on calcule sa corrélation avec le signal s. On récupère alors la fréquence doppler qui maximise la corrélation et on considère que le signal du satellite est présent dans le signal reçu si la valeur obtenue dépasse un certain seuil.

Écrire une routine Matlab permettant, pour un satellite donné, de rechercher sa fréquence doppler avec la procédure ci-dessus. On pourra opérer la recherche avec un pas de discrétisation de 10 Hz sur la fréquence Doppler, *i.e.* Δ sera à optimiser dans un set de valeurs du type :

$$\Delta \in \{-M..., -30, -20, -10, 0, +10, +20, +30, ...M\}$$

où M est la valeur maximale du décalage doppler trouvée à la question Q10.

Appliquer la routine pour trouver les délais de propagation (en ms) de tous les satellites visibles à l'instant de la mesure.

Q14. Noter que si le code se répète tous les 1 ms alors, pour des émetteurs situés à au moins [4] H = 20 000 km, le temps de propagation est supérieur à 66 ms, et donc le code se répète entièrement au moins 66 fois avant d'arriver au récepteur.

On appelle ambigüité de code cette valeur entière, qui doit être déterminée pour chaque satellite en amont du calcul de position. En pratique, un message de navigation modulé à l'intérieur du signal GPS et pourvu de tops de synchronisation, permet de lever cette ambigüité. Ne disposant pas du temps nécessaire dans ce TP pour nous attaquer au décodage de ce message de navigation, nous allons lever l'ambigüité à partir d'une position (très) approximative de la position du récepteur.

Pour ce faire, on prend un point quelconque au centre de la France, par exemple :

$$\lambda = 2.44°, \quad \varphi = 47.00°, \quad h = 0$$

Une conversion de (λ, φ, h) en coordonnées cartésiennes ECEF nous donne :

4. En fonction de la hauteur du satellite dans le ciel par rapport à l'horizon, sa distance peut varier de 20 000 à plus de 30 000 km.

X = 4353736.820 m, Y = 185520.548 m, Z = 4641764.789 m

Ouvrir le fichier d'éphémérides `igu22346_12.sp3` et récupérer les coordonnées du satellite G25 :

X = 14943197.874 m, Y = 5402264.144 m, Z = 21026251.997 m

Il nous suffit alors de calculer la distance entre le point approximatif ci-dessus, et les coordonnées du satellite, modulo la période spatial du signal : $c/1\ ms = 299.792$ km :

```
P = [ 4353736.820  185520.548  4641764.789];  % Coordonnées point approché
S = [14943197.874 5402264.144 21026251.997];  % Coordonnées satellite
N = floor(norm(P-S)/299792);                  % Ambiguité pour G25
```

On trouve alors $N = 67$, ce qui signifie que la valeur de décalage dt trouvée à la question Q11, doit être corrigée de ce nombre entier de millisecondes, et on obtient la mesure de pseudo-distances entre le satellite G25 et le récepteur, corrigée de l'ambiguïté de code :

$$dt = 67 + 0.30225 = 67.30225 \text{ ms}$$

Appliquer la procédure ci-dessus pour déterminer les valeurs d'ambigüités pour tous les satellites en vue.

En pratique, les valeurs trouvées avec cette procédure ne prennent pas en compte l'erreur d'horloge du récepteur (jusqu'à 1 ms) et peuvent donc différer des valeurs vraies d'une unité. Les valeurs vraies des ambigüités sont données dans la sixième colonne du fichier d'éphémérides `igu22346_12.sp3`. Vérifier que les valeurs trouvées avec la procédure simplifiée ci-dessus ne diffère pas de plus d'une unité des valeurs vraies.

Q15. En utilisant les valeurs d'ambigüités données dans la dernière colonne du fichier d'éphémérides, calculer les pseudo-distances *désambiguisées* de tous les satellites en vue à l'instant de la mesure.

IV. Résolution de la position

La résolution des équations de positionnement sortant du cadre de ce TP, nous allons ici simplement former les données nécessaires à la pose du système d'équations, et nous utiliserons l'application en ligne suivante pour résoudre automatiquement le problème :

Résolution du système d'équations GPS
https://raw.githack.com/ymeneroux/amf_wgs84/master/m2sam/appli/index.html

La travail consiste à présent à :

— Corriger les mesures de pseudo-distances pour prendre en compte les erreurs d'horloge des satellites (directement disponibles dans ce même fichier d'éphémérides)

— Passer les mesures de pseudo-distances corrigées et les positions satellites à l'instant de la mesure dans l'application ci-dessus pour estimer la position du récepteur.

Ces deux étapes sont l'objet des deux questions qui suivent (Q16 et Q17).

Q16. Pour corriger les erreurs d'horloge des satellites, on utilise la valeur donnée (en μs) dans la cinquième colonne du fichier d'éphémérides `igu22346_12.sp3`, que l'on ajoute au temps de trajet trouvé à la question Q15 pour chaque satellite. La valeur est alors convertie en pseudo-distance (en mètres) en multipliant par la vitesse de la lumière dans le vide.

Appliquer cette procédure pour déterminer les pseudo-distances de tous les satellites en vue à l'instant de la réception. On s'assurera que le script est correct en vérifiant que le résultat obtenu pour le satellite G25 est proche de 19 993 737 m.

Q17. Utiliser l'application proposée en entête de partie IV pour calculer la position du récepteur. Calculer l'erreur 3D avec la référence donnée en début de l'énoncé de ce TP.

En substance, ce code permet de calculer par moindres carrés la position qui s'ajuste au mieux aux pseudo-distances mesurées entre les satellites et le récepteur à l'instant de l'observation. Formellement, nous avons 4 inconnues (les trois coordonnées d'espace du récepteur, plus une inconnue supplémentaire pour modéliser l'erreur de son horloge interne par rapport au temps du système GPS). Si nous disposons de $n \geqslant 4$ mesures de pseudo-distances vers des satellites de positions connues, le programme cherche la solution la plus juste [5] pour résoudre un système de n équations à 4 inconnues.

Q18. En inspectant les résultats donnés par l'application (bouton `Détails`) calculer (en us) l'erreur de l'horloge du récepteur). Quelle est la précision de la détermination de cette erreur ?

Q19. Utiliser l'application ci-dessous pour exprimer le point en coordonnées géographiques et le représenter sous *Google Maps*.

Conversion de coordonnées
https://tool-online.com/en/coordinate-converter.php

Notons que la précision de positionnement est relativement médiocre. Celà est dû en grande partie au faible taux d'échantillonnage du signal RF reçu. Avec 2.048 Méga-échantillons par seconde, la résolution spatiale du code vue du récepteur est de l'ordre de 150 m. La mesure de corrélation ne peut donc guère descendre en dessous de cette précision. Remarquons cependant que d'autres sources d'erreur devraient également être prises en compte pour améliorer la qualité de positionnement. EN particulier, la traversée de la ionosphère et de la troposphère induit un retard de propagation (vitesse de la lumière réduite et trajectoire courbée) à prendre en compte (erreur de l'ordre de quelques dizaines de mètres).

5. S'il y a strictement plus de 4 satellites en vue, formellement le problème est sur-contraint et n'a pas de solution ; la technique des moindres carrés permet de trouver un compromis optimal entre toutes les observations.