

Développement d'une application de positionnement GNSS sous Android incluant les données GALILEO

Antoine GRENIER

Maitre de stage : Valérie RENAUDIN (GEOLOC)

Référents : Xavier COLLILIEUX (ENSG), Laurent MOREL (ESGT)

GEOLOC/IFSTTAR
(Nantes)

antoine.grenier@ensg.eu

September 16, 2019

Intégration des signaux GALILEO
dans l'application Android *GeolocPVT* pour améliorer le
calcul de position en temps réel.

Plan de la présentation

- 1 GNSS sur Android
- 2 Orbites et horloges précises en temps réel
- 3 Calcul de position du récepteur
- 4 Filtre de Kalman étendu
- 5 Développements futurs

GNSS sur Android

- Depuis 2016, Google API permet récupération des mesures brutes GNSS.
→ Permet le calcul d'une position avec son propre algorithme
- De nombreuses études publiées sur les capacités des récepteurs GNSS sur smartphone.
- En 2018, sortie du Xiaomi Mi8 : 1er téléphone avec récepteur GNSS bi-fréquence (L1/L5)
→ Positionnement Ponctuel Précis (PPP)

Etat initial de *GeolocPVT*

- *GeolocPVT*: application Android développée par le laboratoire Geoloc.
- Calcul de position par moindres carrés mono-fréquence et mono-constellation (GPS)

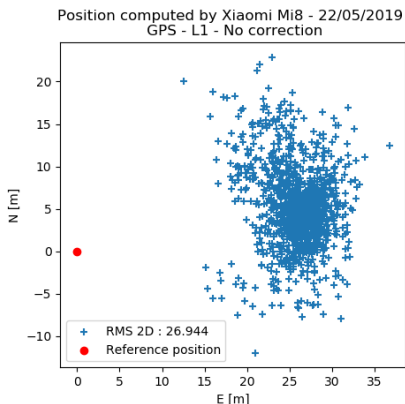


Figure 1: Positions calculées par *GeolocPVT* dans son état initial.

Challenges

- Pseudodistances non récupérables directement pas Google API, seulement les mesures de temps [1]

$$P = \frac{(t_{Rx} - t_{Tx})}{1e9} c \quad (1)$$

- Message de navigation GALILEO non reçu sur téléphone
→ Récupération par internet
- Antenne GNSS low-cost → Mesures de moins bonne qualité

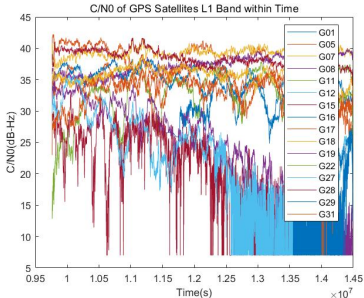


Figure 2: C/n0 observés par Xiaomi Mi8 (Credits: Yazheng Wei, Geoloc).

Orbites et horloges précises en temps réel

- Récupération de position satellite en temps réel
 - SUPL Google - message de navigation
 - IGS ultra-rapid - positions précises
 - RTCM streams (IGS-RTS) - orbites et corrections précises
 - Solution choisie pour GeolocPVT.
- RTCM - Radio Technical Commission for Maritime Services
 - Protocole d'encodage de données GNSS en stream binaire [9]
 - Encodées sous forme de messages pour chaque donnée
 - Corrections précises et multi-constellation
 - Précision aux produits ultra-rapides, échantillonnage des produits horloges satellites à 5 sec
 - Accès et décodage facilité par la bibliothèque *GoGPS*

Récupération dans l'application

- *GoGPS* - Bibliothèque Java pour calcul GPS [8]
 - Possède un module de récupération des streams.
- Intégration dans *GeolocPVT* et adaptation pour Android.
- Décodage messages pour 3 constellations ajoutés : GPS, GALILEO, BEIDOU.
- Récupération en parallèle des calculs de l'application.
 - Permet de toujours avoir les corrections les plus précises possibles au moment de l'acquisition.

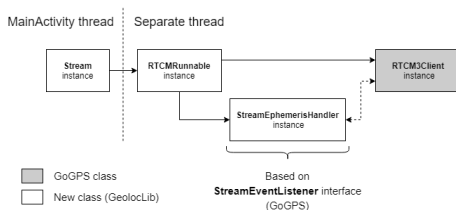


Figure 3: Chaîne de classe pour récupération des streams.

Calcul de position du récepteur

- Calculs d'une position par GNSS basés sur trilatération
 - Distance récepteur-satellite observée et corrigée.
 - 4 satellites au minimum pour résoudre le système $(x, y, z, \delta t_r)$.
 - Pour chaque constellation ajoutées, on estime un biais inter-système.
- Modèle utilisé pour le calcul Standard Point Positioning (SPP) [2, 3]:

$$\begin{aligned} P_f^j &= \rho^j + c \cdot (\delta t_r - \delta t^j) + \tau_{iono}^j + \tau_{tropo}^j + \epsilon \\ \Phi_f^j &= \rho^j + c \cdot (\delta t_r - \delta t^j) - \tau_{iono}^j + \tau_{tropo}^j + \lambda_f \cdot N_f^j + \epsilon \end{aligned} \quad (2)$$

- Lissage des observations de pseudodistances par la phase [2]

$$\hat{p}_k^j = \frac{1}{n} \cdot p_k^j + \frac{n-1}{n} \left(\hat{p}_{k-1}^j + (\Phi_k^j - \Phi_{k-1}^j) \right) \quad \text{if } k < N, n = k \quad (3)$$

else, $n = N$

Corrections

- Correction ionosphérique
 - Xiaomi Mi8 reçoit L1 & L5 → combinaison ionosphere-free
- Correction troposphérique
 - Modèle Saastomanien [2]
- Pondération par C/n_0
 - Qualité des signaux GNSS sur smartphone non dépendante de l'élévation
 - Pondération des signaux sur C/n_0 plus adaptée [7]

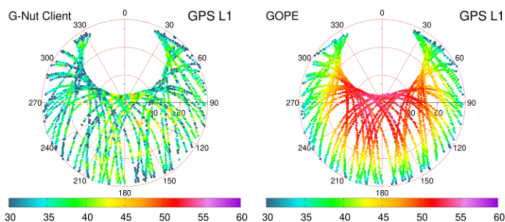


Figure 4: C/n_0 observés par Xiaomi Mi8 et récepteur géodésique [10].

Résultats (1/2)

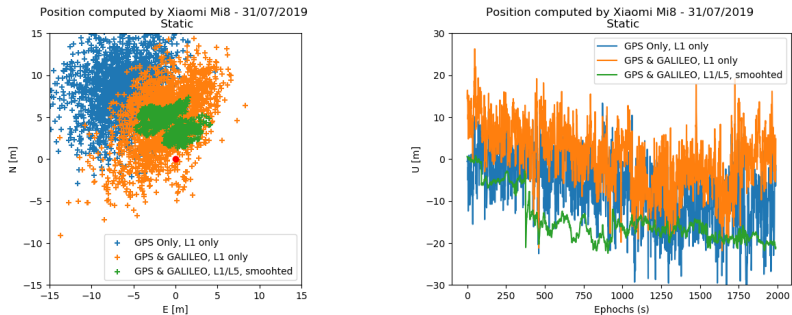
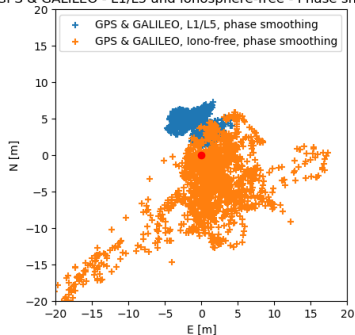


Figure 5: Solution SPP single-epoch, avec et sans lissage de la phase.

Résultats (2/2)

Position computed by Xiaomi Mi8 - 31/07/2019
GPS & GALILEO - L1/L5 and ionosphere-free - Phase smoothing



Position computed by Xiaomi Mi8 - 31/07/2019
GPS & GAL - GDOP

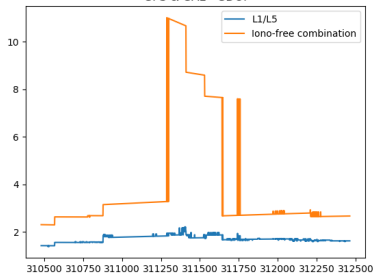


Figure 6: Solution SPP single-epoch, avec mesures L1/L5 ou ionosphere-free.

Filtre de Kalman étendu

- Accumulation des informations sur plusieurs époques
- Modèle de prédiction statique \rightarrow coordonnées fixes [2, 4]
 - Matrice de transition F et matrice de bruit du système Q

$$F = \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & 0 & \\ & & & & 1 \\ & & & & & 1 \end{bmatrix}_{p \times p}$$

$$Q = \begin{bmatrix} 0 & & & & \\ & 0 & & & \\ & & 0 & & \\ & & & \sigma_{cdt_r} & \\ & & & & 0 \\ & & & & & 0 \end{bmatrix}_{p \times p} \quad (4)$$

Mode statique - Résultats

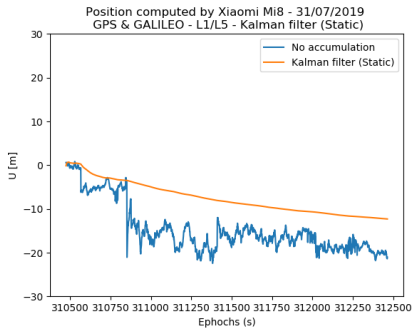
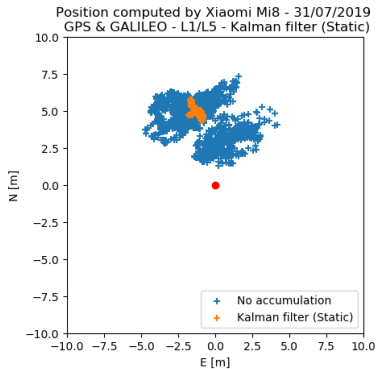


Figure 7: Solution SPP Kalman statique.

Nouveau vecteur d'état avec estimation de vitesse [2, 4, 5]

$$X = [x \quad \dot{x} \quad y \quad \dot{y} \quad z \quad \dot{z} \quad c \cdot \delta t_r \quad c \cdot \delta \dot{t}_r \quad c \cdot GGTO \quad c \cdot GBTO]^T \quad (5)$$

- Estimation de la vitesse par méthode TDCP [11]
 - Time Differenced Carrier Phase

$$\Delta \Phi^j - \Delta D + \Delta g = -e_k \cdot \Delta r_u + c \cdot \Delta \delta \dot{t}_r + \epsilon \quad (6)$$

$$e^j = \frac{u_r - u^s}{\|u_r - u^s\|} = \frac{u_r - u^s}{\rho} \quad (7)$$

$$\Delta g = e_k \cdot r_{u,k-1} - e_{k-1} \cdot r_{u,j-1} \quad (8)$$

$$\Delta D = e_k \cdot r_{s,k} - e_{k-1} \cdot r_{s,j-1} \quad (9)$$

Mode dynamique - Résultats (1/2)

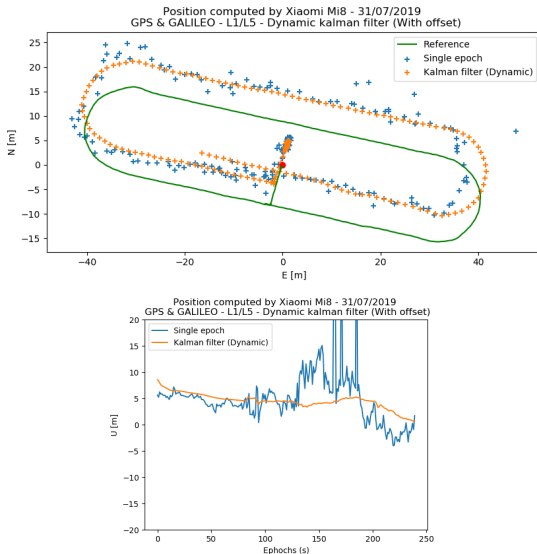


Figure 8: Single-epoch SPP et Kalman dynamique (avec décalage)

Mode dynamique - Résultats (2/2)

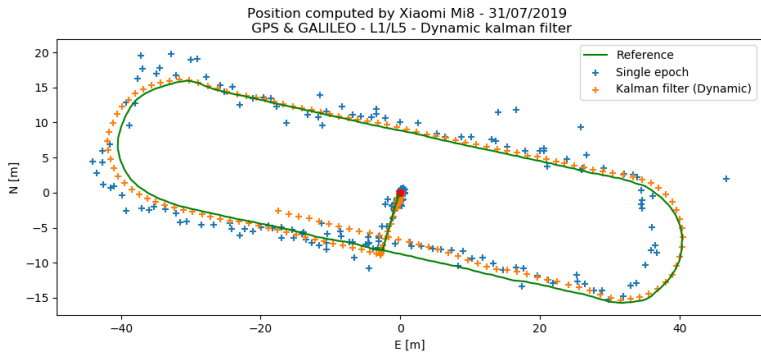


Figure 9: Single-epoch SPP et Kalman dynamique (sans décalage)

Développements futurs

- Amélioration du code (problème décalage coordonnées)
- Intégration de nouveaux messages RTCM (e.g., Satellite Phase Bias)
- Intégration de l'algorithme PPP développé durant le stage

Bilan

Merci pour votre attention.

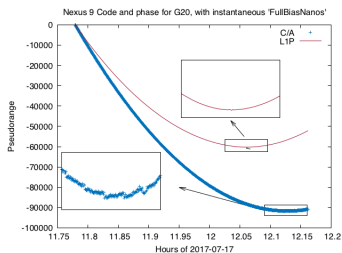
-  European GNSS Agency. *Using GNSS Raw Measurements on Android devices*. 2017. URL: https://www.gsa.europa.eu/system/files/reports/gnss_raw_measurement_web.pdf.
-  European Space Agency. *GNSS Data Processing, Volume I: Fundamentals and Algorithms*. 2013. URL: https://gssc.esa.int/navipedia/GNSS_Book/ESA_GNSS-Book_TM-23_Vol_I.pdf.
-  Pierre Bosser. *Positionnement Ponctuel Précis*. 2018.
-  Jay A. Farrel. *Aided Navigation, GPS with High Rate Sensors*. 2008.
-  The Galfins. *GNSS Compare*. 2018. URL: <https://gnss-compare.readthedocs.io/en/latest/description.html>.
-  Miquel Garcia. *GNSS carrier phase from Nexus 9*. 2017. URL: <https://www.rokubun.cat/gnss-carrier-phase-nexus-9/>.
-  Martin Håkansson et al. "Review of code and phase biases in multi-GNSS positioning". In: (2016).
-  Mirko Reguzzoni and Dominioni Teruzzi. *GoGPS*. Geomatics Laboratory of Politecnico di Milano, Italy. URL: https://github.com/goGPS-Project/goGPS_Java.
-  RTCM. *RTCM 10403.3, Differential GNSS (Global Navigation Satellite Systems) Services*. Radio Technical Commission For Maritime Services. 2016. URL: <http://www.rtc.org/differential-global-navigation-satellite--dgnss--standards.html>.
-  Pavel Vaclavovic. *Analyses of raw GNSS observations collected by the Android device Xiaomi Mi8*. RIGTC, 2019.
-  Feng Zhu et al. "Walker: Continuous and Precise Navigation by Fusing GNSS and MEMS in Smartphone Chipset for Pedestrians". In: *Remote sensing* (2019). URL: <https://doi.org/10.3390/rs11020139>.

Mesures GNSS brutes sur Android

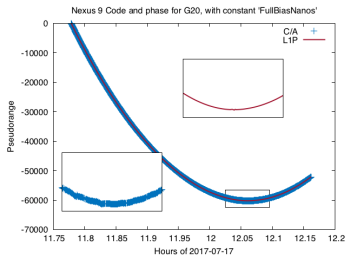
- Pseudodistances non récupérables directement pas Google API, seulement les mesures de temps [1]

$$P = \frac{(t_{Rx} - t_{Tx})c}{1e9} \quad (10)$$

- Temps de réceptions t_{Rx} contient des biais qui doivent être fixés pour éviter des divergences entre mesures;



Range and phase using instantaneous `FullBiasNanos`



Using constant `FullBiasNanos` mitigates the code/phase divergence due to the receiver clock

Figure 10: Generation pseudodistances avec et sans fixer le biais *Fullbias* (Source: Miquel Garcia [6])

- Linéarisation sous la forme $Y = HX$ par développement de Taylor
 - X le vecteur d'état

$$X = [dx \quad dy \quad dz \quad c \cdot dt_r \quad c \cdot GGTO \quad c \cdot GBTO]^T \quad (11)$$

- Y le vecteur d'observation

$$Y = \begin{bmatrix} \vdots \\ \rho_i^{j,GAL} - \rho^j{}^{,GAL} - c \cdot \delta t_r + c \cdot \delta t^{j,GAL} - c \cdot GGTO \\ \vdots \end{bmatrix}_{n \times 1} \quad (12)$$

- H la matrice d'observation

$$H = \begin{bmatrix} \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{x_r - x^j{}^{,GAL}}{\rho^j{}^{,GAL}} & \frac{y_r - y^j{}^{,GAL}}{\rho^j{}^{,GAL}} & \frac{z_r - z^j{}^{,GAL}}{\rho^j{}^{,GAL}} & 1 & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}_{n \times p} \quad (13)$$

