

Algorithmes de positionnement par fusion de données GNSS

Contenu

5.1.	Introduction.....	102
5.2.	Méthodes des moindres carrés itérativement repondérés.....	102
5.2.1.	Théorie de MM-Estimation	104
5.2.2.	Etapes de l’algorithme d’IRWLS	105
5.2.3.	Présentation de l’algorithme RAIM.....	107
5.2.4.	Résultats de simulation.....	109
5.3.	Filtrage de Kalman étendu robuste	111
5.3.1.	Modèle de filtrage de Kalman étendu robuste.....	111
5.3.2.	Application de l’algorithme FKE-robuste	113
5.3.3.	L’algorithme IRWLS	119
5.3.4.	Résultats de simulation.....	120
5.4.	Combinaison des algorithmes FKE-robuste et LSTM (Long Short-Term Memory)	121
5.4.1.	Application de la méthode FKE-robuste pour les données issues des trois satellites (GPS, Galileo, GLONASS)	122
5.4.1.1.	Application de la méthode FKE-robuste	122
5.4.1.2.	Résultats de simulation.....	126
5.4.2.	Méthode de dé-bruitage de filtre	130
5.4.2.1.	Modèle de dé-bruitage de filtre	131
5.4.2.2.	Résultats de simulation.....	135
5.5.	Positionnement relatif	137
5.5.1.	Résultats de simulation.....	137
5.6.	Conclusion	141

Ce chapitre présente les algorithmes de fusion de données GNSS utilisés pour le positionnement. Nous présentons également quelques techniques pour éliminer des données aberrantes. Dans un premier temps, nous présentons la méthode des « moindres carrés itérativement repondérés ». Ensuite, nous présentons la méthode du « filtre de Kalman étendu robuste ». Puis, nous présentons la méthode LSTM (Long Short Term Memory) pour prédéterminer les erreurs de comportement du filtrage de Kalman robuste étendu. Nous finirons ce chapitre avec la fusion de données GNSS en appliquant le positionnement relatif présenté dans le chapitre 4.

5.1. Introduction

Récemment, les multi-constellations et les multi-fréquences ont prouvé qu'elles pouvaient améliorer la précision du positionnement. Ces progrès ont entraîné une croissance significative du nombre d'applications GNSS dans de nouveaux domaines et services : véhicules sans pilote, la défense, construction, agriculture de précision, ... L'utilisation de signaux multiples présente des avantages mais également des inconvénients. En effet, ils peuvent contenir des signaux de mauvaise qualité et qui ont un impact négatif sur la précision de position. C'est pourquoi, nous présenterons dans ce chapitre deux algorithmes : celui des moindres carrés itérativement repondérés et le filtre de Kalman étendu robuste pour éliminer les données aberrantes. Ces algorithmes sont appliqués pour le mode de positionnement absolu. En outre, on va présenter aussi une autre méthode, basée sur le « deep learning », combinée avec l'algorithme du filtre de Kalman étendu robuste afin d'améliorer la précision de position.

Comme indiqué dans le chapitre 2, le positionnement relatif est un autre mode de positionnement plus précis car il permet d'éliminer les erreurs systématiques en réalisant des différences d'observation entre des stations. Pour cela, il est nécessaire de disposer d'au moins une station dont la position est connue. Dans ce chapitre, ce positionnement est réalisé en se basant sur la fusion des données GPS et GLONASS, on utilise ensuite un filtre de Kalman étendu pour déterminer la position.

5.2. Méthodes des moindres carrés itérativement repondérés

La méthode des moindres carrés itérativement repondérés (Iterative ReWeighted Least Squares – IRWLS) ([47], [48]) est utilisée pour optimiser les résultats par une méthode itérative dans laquelle chaque étape consiste à résoudre un problème des moindres carrés pondérés.

Cet algorithme est appliqué en combinant des données GPS et GLONASS. La position résultante des deux systèmes sera exprimée dans le même système de coordonnées et le même système de temps. Cependant, les coordonnées de GPS et GLONASS sont exprimées respectivement en coordonnées WGS-84 et PZ-90.11 (Cf. la section 2.2). Bien qu'ils soient différents, l'écart dans les valeurs des coordonnées est de l'ordre du centimètre. On peut donc considérer que les deux coordonnées sont identiques. Quant à la référence de temps, le temps GPS (GPST) a été synchronisé à l'UTC le 6 janvier 1980, et il évolue de manière continue. Il n'insère aucune seconde de recalage contrairement à l'UTC. Concernant le temps GLONASS (GLONASST), il se base sur l'UTC ou plus précisément, GLONASST est lié à l'UTC (SU). Ce n'est donc pas un temps continu car des recalages peuvent intervenir. La différence entre les deux systèmes de temps est donnée par l'équation (5.1).

$$GPST - GLONASST = \Delta tLS + \tau GPS \quad (5.1)$$

Où ΔtLS correspond aux secondes intercalaires entre GPST et UTC ; τGPS est le décalage de temps fractionnaire entre les deux systèmes (GPS et GLONASS).

Après synchronisation des systèmes de coordonnées et de temps, la démarche adoptée pour l'estimation de la position est présentée ci-dessous.

Après avoir éliminé les erreurs GNSS : de l'horloge des satellites, ionosphériques, troposphériques et multi-trajets, les pseudo-distances corrigées des deux systèmes peuvent s'écrire [46] :

$$\rho_c^{G,R} = \sqrt{(x_{G,R} - x_u)^2 + (y_{G,R} - y_u)^2 + (z_{G,R} - z_u)^2} + c\delta t_{G,R} + \varepsilon_{\rho,G,R} \quad (5.2)$$

Où : les exposants et indices G, R se réfèrent respectivement aux satellites GPS et GLONASS ; ρ_c est la mesure de la pseudo-distance corrigée ; c est la vitesse de lumière ; $[x_{G,R}, y_{G,R}, z_{G,R}]$ sont les positions des satellites GPS et GLONASS ; $[x_u, y_u, z_u]$ est la position de l'utilisateur ; δt est le décalage d'horloge du récepteur ; ε_ρ est le bruit de la mesure.

Combiné les deux systèmes signifie la résolution d'un système à 5 inconnues (x_u, y_u, z_u : position du récepteur, δt_G et δt_R : les décalages d'horloge du récepteur avec le temps GPS et le temps GLONASS). Alors, le vecteur d'état est défini par :

$$X = [x_u \ y_u \ z_u \ c\delta t_G \ c\delta t_R] \quad (5.3)$$

L'équation (5.2) peut être écrite sous forme compacte comme :

$$\rho = \mathbf{G}X + \varepsilon \quad (5.4)$$

ρ représente les pseudo-distances corrigées de GPS et GLONASS ; G est la matrice de géométrie, elle est donnée par l'équation (5.5).

$$G = \begin{bmatrix} a_x^{GPS,1} & a_y^{GPS,1} & a_z^{GPS,1} & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_x^{GPS,N_{gps}} & a_y^{GPS,N_{gps}} & a_z^{GPS,N_{gps}} & 1 & 0 \\ a_x^{GLO,1} & a_y^{GLO,1} & a_z^{GLO,1} & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_x^{GLO,N_{glo}} & a_y^{GLO,N_{glo}} & a_z^{GLO,N_{glo}} & 0 & 1 \end{bmatrix} \quad (5.5)$$

Avec :

$$\begin{aligned} a_x^i &= \frac{-(x_i - x_u)}{\sqrt{(x_i - x_u)^2 + (y_i - y_u)^2 + (z_i - z_u)^2}} \\ a_y^i &= \frac{-(y_i - y_u)}{\sqrt{(x_i - x_u)^2 + (y_i - y_u)^2 + (z_i - z_u)^2}} \\ a_z^i &= \frac{-(z_i - z_u)}{\sqrt{(x_i - x_u)^2 + (y_i - y_u)^2 + (z_i - z_u)^2}} \end{aligned} \quad (5.6)$$

Pour $i = 1, 2, \dots, N_{gps} + N_{glo}$, où N_{gps} est le nombre de satellites GPS suivis ; N_{glo} est le nombre de satellites GLONASS suivis ; $[x_i, y_i, z_i]$ est la position du $i^{\text{ème}}$ satellite et $[x_u, y_u, z_u]$ est la position estimée de l'utilisateur.

On suppose \hat{X} le vecteur d'état estimé. Alors, $\delta\rho$ est la différence entre les pseudo-distances d'estimés $\hat{\rho}$ et les pseudo-distances corrigées $\tilde{\rho}$ des deux systèmes GPS et GLONASS. δX est la différence entre le vecteur d'état d'estimé final \hat{X} et le vecteur d'état estimé initial \tilde{X} . L'équation (5.4) peut être écrite comme :

$$\delta\rho = \mathbf{G}\delta X + \varepsilon \quad (5.7)$$

Un algorithme des moindres carrés itératifs repondérés (IRWLS) est utilisé pour résoudre le système. La solution de l'équation (5.7) peut être réécrite comme :

$$\delta X = (G^T W G)^{-1} G^T W \delta \rho \quad (5.8)$$

La matrice des poids W de la méthode IRWLS est calculée par la méthode « MM-Estimation », cette méthode fait l'objet de la section suivante.

5.2.1. Théorie de MM-Estimation

Dans l'analyse de régression, l'utilisation de la méthode des moindres carrés ne serait pas appropriée pour résoudre un problème contenant des observations aberrantes ou extrêmes. Nous avons donc besoin d'une méthode de stimulation des paramètres et qui soit robuste. Plusieurs types d'estimation existent [49] : M-Estimation, S-Estimation, MM-Estimation, ... en régression robuste pour déterminer un modèle de régression. M-Estimation est une extension de la méthode d'estimation du maximum de vraisemblance ([50],[51]). S-Estimation est basée sur l'échelle résiduelle de M-Estimation. S-Estimation est proposée par Rousseeuw et Yohai [52]. La procédure MM-Estimation consiste à estimer le paramètre de régression à l'aide de S-Estimation qui minimise l'échelle de résidu à partir de M-Estimation. MM-Estimation est proposée par Yohai [53]. MM-Estimation est connue comme une classe de statistiques robustes pour une analyse des données statistiques plus efficace avec un point de rupture plus élevé (high breakpoint). Le point de rupture élevé est la mesure de proportion des valeurs aberrantes qui peuvent être traitées efficacement sans détériorer les résultats. Dans ce mémoire, on retient donc MM-Estimation [48] pour calculer la matrice des poids.

On suppose que l'on a une équation linéaire générale de la forme :

$$Y = GX + \epsilon \quad (5.9)$$

Où Y est le vecteur d'observation ; G est le vecteur de transformation pour l'observation, X est le vecteur d'état inconnu et ϵ est le vecteur de bruit correspondant à Y .

La méthode robuste est utilisée pour limiter l'influence des valeurs aberrantes sur les estimations de l'état X . L'objectif de l'estimateur robuste est de rechercher un critère d'ajustement optimal aux observations réelles afin de réduire l'effet des données anormales. En général, un estimateur robuste minimise la fonction de coût $J(r)$ donné par :

$$J(r) = \sum_{i=1}^n \mu(r_i) = \sum_{i=1}^n \mu(y_i - g_i X) \quad (5.10)$$

Où n est le nombre d'observations ; y_i est la $i^{\text{ème}}$ observation ($i^{\text{ème}}$ ligne de Y) ; g_i est le vecteur de transformation pour la $i^{\text{ème}}$ observation ($i^{\text{ème}}$ ligne de G), et $r_i = y_i - g_i X$ est l'erreur résiduelle pour la $i^{\text{ème}}$ observation. $\mu(r)$ est la fonction objectif qui donne la contribution de chaque résidu à la fonction de coût $J(r)$. L'estimation échelonnée est utilisée pour minimiser dynamiquement le poids des observations hors contraintes :

$$\min \sum_{i=1}^n \mu\left(\frac{r_i}{s}\right) = \sum_{i=1}^n \mu\left(\frac{y_i - g_i X}{s}\right) \quad (5.11)$$

Où s est le facteur d'échelle. $\check{r}_{s,i}$ est le résiduel mis à l'échelle $\frac{r_i}{s}$. Si φ est la dérivée de la fonction objective pour les inconnues X , alors $\varphi = \mu'|_X$. Différencier la fonction objective dans l'équation (5.11) revient à prendre la dérivée égale à zéro,

$$-\frac{1}{s} \sum_{i=1}^n \varphi\left(\frac{y_i - g_i X}{s}\right) g_i = 0 \quad (5.12)$$

Soit:

$$\sum_{i=1}^n \varphi\left(\frac{y_i - g_i X}{s}\right) g_i = 0 \quad (5.13)$$

En introduisant une fonction de pondération $w(\check{r}_{s,i})$ pour considérer les effets de chaque résiduel mis à l'échelle $\check{r}_{s,i}$, l'Eq. (5.13) peut être modifié comme suit :

$$\sum_{i=1}^n w(\check{r}_{s,i}) \varphi(\check{r}_{s,i}) g_i = 0 \quad (5.14)$$

La solution de l'équation (5.14) est une estimation des moindres carrés pondérés. Cependant, les poids sont déterminés par les résidus échelonnés qui dépendent des coefficients estimés, eux-mêmes dépendant de la pondération. Pour cette raison, les moindres carrés itératifs repondérés sont utilisés pour optimiser la solution. La fonction de pondération joue un rôle important et certaines fonctions possibles sont présentées dans le Tableau 5.1 :

Tableau 5.1 : Fonctions de poids

Nom de la méthode	Fonction de poids
Huber[48]	$w(\check{r}_{s,i}) = \begin{cases} 1 & \check{r}_{s,i} \leq \alpha \\ \alpha/ \check{r}_{s,i} & \check{r}_{s,i} > \alpha \end{cases}$
Bi-Tukey[48]	$w(\check{r}_{s,i}) = \begin{cases} 1 & \check{r}_{s,i} \leq \alpha \\ 0 & \check{r}_{s,i} > \alpha \end{cases}$
Bi-Square[48]	$w(\check{r}_{s,i}) = \begin{cases} \left[1 - \left(\check{r}_{s,i} /\alpha\right)\right]^2 & \check{r}_{s,i} \leq \alpha \\ 0 & \check{r}_{s,i} > \alpha \end{cases}$
IQR ⁽¹⁾ [54]	$w(\check{r}_{s,i}) = \begin{cases} 0 & \check{r}_{s,i} < MIN \text{ ou } \check{r}_{s,i} > MAX \\ 1 & \text{autres} \end{cases}$

(1) IQR (InterQuartile Range) ; Q_1 et Q_3 : les valeurs médianes dans la première et la deuxième moitié de l'ensemble de données classées. $MIN = Q_1(\check{r}_s) - m_{IQR}$; $MAX = Q_3(\check{r}_s) + m_{IQR}$; $m_{IQR} = 1.5 * IQR(\check{r}_s)$; $IQR(\check{r}_s) = Q_3 - Q_1$.

Dans le Tableau 5.1, $\check{r}_{s,i}$ est le résidu de la $i^{\text{ème}}$ observation mis à l'échelle. α est une constante de réglage avec une valeur optimale pour atteindre plus de résistance aux valeurs aberrantes (le point de rupture le plus élevé) ainsi que l'efficacité de convergence.

5.2.2. Etapes de l'algorithme d'IRWLS

Après avoir présenté la méthode MM-Estimation, nous appliquerons la méthode des moindres carrés itératifs repondérés (IRWLS) pour résoudre l'équation (5.7) $\delta\rho = \mathbf{G}\delta X + \varepsilon$. L'IRWLS est effectuée pour éliminer les valeurs aberrantes. Un aperçu de la procédure d'estimation robuste est défini dans les étapes suivantes [48] :

- 1) Calculer la matrice diagonale pondérée initiale W_0 . $\delta\rho$ peut impacter les résultats. Alors, il est nécessaire d'éliminer les données aberrantes de $\delta\rho$ à l'aide de la fonction de poids Bi-Tukey :

$$W_0(i, i) = \begin{cases} 1 & \text{if } |\bar{\delta\rho}(i)| \leq \alpha \\ 0 & \text{if } |\bar{\delta\rho}(i)| > \alpha \end{cases} \quad (5.15)$$

Où : $\bar{\delta\rho} = \frac{\delta\rho}{1.4826 \times \text{median}(|\delta\rho|)}$ avec $i = 1, 2, \dots, n$; n est le nombre d'observations et α est une constante.

- 2) Trouver une correction δX_1 :

$$\delta X_1 = (G_1^T W_0 G_1)^{-1} G_1^T W_0 \delta\rho \quad (5.16)$$

- 3) Mettre à jour le vecteur d'état \hat{X}_1

$$\hat{X}_1 = \hat{X}_0 + \delta X_1 \quad (5.17)$$

- 4) Les résidus sont estimés :

$$r_1 = \delta\rho - G_1^T \delta X_1 \quad (5.18)$$

- 5) Plusieurs fonctions d'estimation d'échelle sont définies, mais la fonction médiane suivante est utilisée pour calculer la valeur d'échelle initiale ([48], [55], [56]):

$$s_1 = 1.4826 * \text{median}(|r_1|) \quad (5.19)$$

- 6) Mettre à jour la matrice des poids initiale à l'aide de Bi-Tukey :

$$W_1(i, i) = \begin{cases} 1 & \left| \frac{r_1(i)}{s_1} \right| \leq \alpha \\ 0 & \left| \frac{r_1(i)}{s_1} \right| > \alpha \end{cases} \quad (5.20)$$

- 7) Boucle (j est la j^{ème} itération)

- a) Mettre à jour la valeur de la matrice G_j avec X_{j-1}
- b) Calculer la correction δX_j

$$\delta X_j = (G_j^T W_{j-1} G_j)^{-1} G_j^T W_{j-1} \delta\rho \quad (5.21)$$

- c) Mettre à jour le vecteur d'état

$$\hat{X}_j = \hat{X}_{j-1} + \delta X_j \quad (5.22)$$

- d) Calculer HPL (Horizontal Protect Level)

e) Si $\|\delta X_j\| \leq 0.001$; break

f) Si $\|\delta X_j\| > 0.001$; continue

- g) Mettre à jour les résidus estimés

$$r_j = \delta\rho - G_j^T \delta X_j \quad (5.23)$$

- h) Calculer IQR

$$IQR(r_j) = Q_3(r_j) - Q_1(r_j) \quad (5.24)$$

Q_1 et Q_3 sont les valeurs moyennes dans la première moitié et dans la deuxième moitié de l'ensemble des données classées.

i) Trouver les seuils intérieurs

$$MIN = Q_1(r_j) - 1.5 * IQR(r_j) \quad (5.25)$$

$$MAX = Q_3(r_j) + 1.5 * IQR(r_j)$$

j) Mettre à jour la matrice pondérée

$$W_j(i, i) = \begin{cases} 0 & r_j(i) < MIN \text{ or } r_j(i) > MAX \\ W_{j-1}(i, i) & \text{autres} \end{cases} \quad (5.26)$$

k) Aller à l'étape a)

8) Fin boucle

9) Si $HPL \leq HAL$, les positions estimées sont alors acceptées (HAL : Horizontal Alert Limit)

Si non, les positions estimées ne sont pas acceptées.

Le calcul de HPL est effectué par l'algorithme RAIM présenté dans la section ci-dessous 5.2.3.

5.2.3. Présentation de l'algorithme RAIM

L'algorithme RAIM (Receiver Autonomous Integrity Monitoring) est utilisé pour détecter les erreurs de positionnement dépassant la limite d'alerte. L'un des résultats de RAIM est le niveau de protection horizontale (HPL). Le HPL est le rayon d'un cercle dans le plan horizontal, centré à la position réelle. La limite d'alerte horizontale (HAL) est le HPL maximum autorisé. Cela signifie que les positions estimées sont acceptées lorsque $HPL \leq HAL$. Les étapes de calcul du paramètre HPL sont présentées ci-dessous.

Considérons un modèle linéaire de mesures d'observations qui peut être écrit comme l'équation (5.9) : $Y = GX + \epsilon$

Le vecteur d'état de navigation est estimé par l'estimateur des moindres carrés pondérés :

$$\hat{X} = (G^T W G)^{-1} G^T W Y \quad (5.27)$$

Le vecteur d'erreur d'état est défini comme suit :

$$\begin{aligned} \delta X &= \hat{X} - X = (G^T W G)^{-1} G^T W Y - X = (G^T W G)^{-1} G^T W (GX + \epsilon) - X \\ &= (G^T W G)^{-1} G^T W \epsilon \end{aligned} \quad (5.28)$$

Le résidu estimé $\hat{\epsilon}$ est déterminé par la différence entre l'observation mesurées y et les observations estimées :

$$\begin{aligned} \hat{\epsilon} &= Y - G\hat{X} = Y - (G^T W G)^{-1} G^T W Y = (I - (G^T W G)^{-1} G^T W) Y = S Y \\ &= [I - G(G^T W G)^{-1} G^T W] [GX + \epsilon] \\ &= GX + \epsilon - G(G^T W G)^{-1} G^T W GX - G(G^T W G)^{-1} G^T W \epsilon \\ &= GX + \epsilon - GX - G(G^T W G)^{-1} G^T W \epsilon \\ &= [I - G(G^T W G)^{-1} G^T W] \epsilon = S \epsilon \end{aligned} \quad (5.29)$$

Où

$$S = I - (G^T W G)^{-1} G^T W \quad (5.30)$$

L'écart-type a priori de l'observation est calculé par :

$$\sigma_0 = \sqrt{\hat{\epsilon}^T W \hat{\epsilon} / \sqrt{n - p}} \quad (5.31)$$

Où : n est le nombre d'observations et p est le nombre de paramètres à l'état inconnu de X .

L'erreur de position dans le vecteur d'erreur d'état δX (Eq. (5.28)) est définie dans le système (x, y, z) . Dans la pratique, un système de coordonnées local (E, N, U) est plus approprié. Pour transformer le système de coordonnées (x, y, z) en coordonnées (E, N, U), une transformation est nécessaire, ainsi une approche simplifiée consiste à utiliser la matrice de transformation orthogonale F^T :

$$F^T = \begin{bmatrix} -\sin \lambda & \cos \lambda & 0 \\ -\sin \varphi \cos \lambda & -\sin \varphi \sin \lambda & \cos \varphi \\ \cos \varphi \cos \lambda & \cos \varphi \sin \lambda & \sin \varphi \end{bmatrix} \quad (5.32)$$

Où φ est la longitude et λ est la latitude.

Ici, seuls les paramètres des coordonnées sont pris en compte. Par exemple, le vecteur d'état X est défini par l'Eq. (5.3) avec la première, deuxième et troisième position par rapport à trois (x, y, z) coordonnées. Par conséquent, la nouvelle matrice G_0 est obtenue sous la forme $G_0 = G(:, 1:2:5)$ et les erreurs de position sont $\delta X_0 = \delta X(1:2:5)$. La matrice M relative à l'Est, au Nord et vers le haut (Up) est calculée comme suit :

$$\delta X_{ENU} = \begin{bmatrix} E \\ N \\ U \end{bmatrix} = F^T \delta X_0 = F^T (G_0^T G_0)^{-1} G_0^T \epsilon = M \epsilon \quad (5.33)$$

L'erreur horizontale concerne l'abscisse et le nord est pris en compte. Par conséquent, la matrice M_0 peut être définie comme $M_0 = M(1:2,:)$.

Imaginez maintenant une défaillance de magnitude β du $i^{\text{ème}}$ satellite (β est placé comme la $i^{\text{ème}}$ composante) :

$$\epsilon = [0 \quad \dots \quad \beta \quad \dots \quad 0]^T \quad (5.34)$$

La norme δX_{EN} (les erreurs de position dans l'axe Est et Nord) au carré pour ce choix spécial de ϵ est donnée par :

$$\|\delta X_{EN}\|^2 = \epsilon^T \tilde{M}_0^T \tilde{M}_0 \epsilon = (m_{1i}^2 + m_{2i}^2) \beta^2 \quad (5.35)$$

De l'Eq. (5.29), $\hat{\epsilon} = S \epsilon$ est rappelé. On a $S^T S = S$ ce qui mène à :

$$\|\hat{\epsilon}\|^2 = \hat{\epsilon}^T \hat{\epsilon} = \epsilon^T S^T S \epsilon = s_{ii} \beta^2 \quad (5.36)$$

L'entrée diagonale (i, i) de S est appelée s_{ii} . Maintenant, à partir de l'Eq. (5.4) et l'Eq. (5.5) :

$$\|\delta X_{EN}\|^2 = \frac{m_{1i}^2 + m_{2i}^2}{s_{ii}} \|\hat{\epsilon}\|^2 \quad (5.37)$$

Où la position d'utilisateur dans le vecteur d'état X est la première, la deuxième et la troisième composante de X . Par conséquent, la nouvelle matrice \tilde{G} est obtenue sous la forme $\tilde{G} =$

$G(:,1:3)$ et les erreurs de position $\delta\tilde{X} = \delta X(1:3)$. La matrice \tilde{M} relative à l'Est, au Nord et vers le haut (Up) est calculée comme suit :

$$\delta X_{ENU} = \begin{bmatrix} E \\ N \\ U \end{bmatrix} = F^T \delta\tilde{X} = F^T (\tilde{G}^T \tilde{G})^{-1} \tilde{G}^T \epsilon = \tilde{M} \epsilon \quad (5.38)$$

L'erreur horizontale concerne l'Est et le Nord. Par conséquent, la matrice \tilde{M}_0 peut être définie comme $\tilde{M}_0 = \tilde{M}(1:2, :)$.

On suppose maintenant une défaillance de magnitude β de la $i^{\text{ème}}$ observation (β est placé comme la $i^{\text{ème}}$ composante) :

$$\|\delta X_{EN}\| = \sqrt{\frac{m_{1i}^2 + m_{2i}^2}{s_{ii}}} \|\hat{\epsilon}\| = \alpha_i \|\hat{\epsilon}\| \quad (5.39)$$

Cette équation est la fonction linéaire avec une ligne droite passant par l'origine et dont la pente est α_i . La pente α_i de l'axe du mode de défaillance au $i^{\text{ème}}$ satellite est calculée pour $i = 1, \dots, n$:

$$\alpha_i = \sqrt{\frac{m_{1i}^2 + m_{2i}^2}{s_{ii}}} \quad (5.40)$$

L'axe de mode avec le plus grand α_i est désigné par α_{max} et le niveau de protection horizontale (HPL) est donné par :

$$HPL = \alpha_{max} \sigma_0 \quad (5.41)$$

Où σ_0 est calculé en utilisant l'Eq. (5.31).

5.2.4. Résultats de simulation

Pour appliquer l'algorithme des « moindres carrés itérativement repondérés », nous avons utilisé les données GPS et les données GLONASS de la station ABMF au format RINEX 2.11. Cette station de référence est située en Guadeloupe. Les données RINEX et les éphémérides de diffusion utilisées pour le calcul ont été téléchargées depuis le serveur CDDIS [39] et le réseau RGP [40] pour le premier janvier 2018.

Les données entre 15h00 et 24h00 sont utilisées pour vérifier les performances d'une estimation robuste. Pour faciliter la comparaison et l'analyse, trois scénarios sont adoptés :

Scénario #1 : solution de navigation basée sur les données GPS

Scénario #2 : solution de navigation basée sur les données GPS / GLONASS

Scénario #3 : solution de navigation robuste basée sur les données GPS / GLONASS.

Dans le scénario #1, on utilise les données GPS avec la méthode des « moindres carrés » pour calculer la position du récepteur. Dans le scénario #2, on utilise la combinaison des données GPS et des données GLONASS avec la méthode des « moindres carrés ». Dans le scénario #3, on utilise la combinaison des données GPS et GLONASS par la méthode « IRWLS » pour améliorer la précision de position de l'utilisateur.

Les résultats de simulation des trois scénarios sont présentés dans la Figure 5.1.

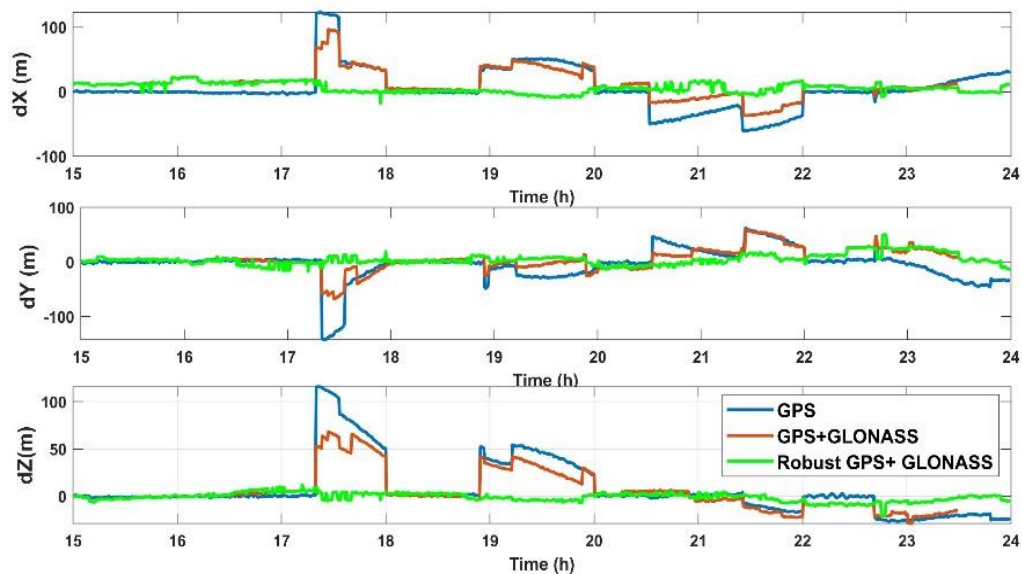


Figure 5.1 : Erreur de position estimée en considérant les trois scénarios.

Tableau 5.2 : Position d'erreur RMS

Erreur RMS (m)	Scénario #		
	1	2	3
RMS_X	32.1	24.9	9.5
RMS_Y	29.0	19.9	11.0
RMS_Z	29.2	20.8	4.4
3D-RMS	52.2	38.0	15.2

Tableau 5.3 : Erreur Max, Min et l'écart-type

Erreur (m)	Scénario #		
	1	2	3
Max_X	123.0	96.6	23.1
Min_X	5e-03	0.2	13e-04
Ecart type X	25.4	16.5	5.9
Max_Y	142.6	67.5	50.5
Min_Y	38e-04	24e-03	19e-04
Ecart type Y	24.0	14.5	7.8
Max_Z	115.8	68.2	20.5
Min_Z	3.6e-4	65e-04	5e-03
Ecart type	24.5	16.0	2.7

Dans la Figure 5.1, la courbe en bleu représente le scénario #1 basé sur les données GPS. On peut noter qu'il y a des défauts évidents vers 17h00 - 18h00 et 19h00 - 22h00. L'erreur de position est supérieure à 100 (m) à 17h00-18h00 et à environ 50 (m) à 19h00 - 22h00. La précision de position dans le Tableau 5.2 montre que la 3D-RMS du scénario #1 est de 52.21 (m), ce qui est grand. La courbe en rouge représente le scénario # 2 basé sur des données GPS / GLONASS. Elle montre que la précision de position du système double GPS / GLONASS est mieux que celle du système simple GPS. Son 3D-RMS est de 38.08 (m), ce qui est toujours grand. Par conséquent, l'estimation robuste est utilisée, la plupart des observations de défauts sont exclues avec succès dans le scénario # 3 (courbe verte). On peut voir que les positions de précision sont considérablement améliorées. Son 3D-RMS est de 15.2 (m). Il est amélioré de 71.0% par rapport au cas où seules les données GPS sont utilisées (52.2%).

5.3. Filtrage de Kalman étendu robuste

Le filtre de Kalman étendu est une extension du filtre de Kalman pour résoudre les problèmes de filtrage optimal non-linéaire. Il est utilisé dans plusieurs domaines technologiques (radar, vision électrique, communication, ...). Cependant, dans le positionnement hybride, les grandes valeurs aberrantes peuvent avoir un impact négatif sur la précision de la position. Les valeurs aberrantes du filtre de Kalman qui en résultent peuvent être classées en trois types : observation, innovation et structure [9][57]. Dans le travail réalisé et présenté dans ce mémoire, un filtre de Kalman étendu robuste est présenté pour traiter les observations et les innovations aberrantes. Il s'agit d'une combinaison du filtre de Kalman étendu avec l'algorithme des moindres carrés itératifs repondérés et la matrice de poids définie par MM-Estimation (section 5.2.1) pour éliminer les valeurs aberrantes. Alors que sa combinaison avec l'algorithme RAIM (5.2.3) est utilisée pour vérifier la position exacte de l'utilisateur.

5.3.1. Modèle de filtrage de Kalman étendu robuste

Le modèle de filtrage de Kalman étendu peut être défini par les deux équations suivantes :

Equation d'état :

$$X_k = f(X_{k-1}) + \epsilon_k \quad (5.42)$$

Equation d'observation :

$$Z_k = h(X_k) + e_k \quad (5.43)$$

Où : X_k est le vecteur d'état et Z_k est la mesure à l'instant t_k ; ϵ_k est le vecteur qui transmet les sources d'erreur système ($\epsilon_k \sim N(0, Q_k)$) ; e_k est le vecteur qui représente les sources d'erreur de mesure $e_k \sim N(0, R_k)$; $f()$ est la fonction pour la transition d'état et $h()$ est la fonction pour la mesure.

Le FKE-robuste combine le filtre de Kalman étendu avec les moindres carrés itératifs repondérés (IRWLS). L'IRWLS travaille sur la régression linéaire pour traiter le problème des valeurs aberrantes dans les données. Par conséquent, la conversion d'un modèle non-linéaire en un modèle linéaire est nécessaire pour appliquer le FKE-robuste. Alors, le FKE-robuste se déroule en trois étapes : linéarisation des équations, reformulation du filtre et mise à jour.

Etape 1 : Linéarisation des équations

La linéarisation de la fonction $f(X_{k-1})$ dans l'Eq. (5.42) :

$$A_k = \left. \frac{\partial f}{\partial X} \right|_{X=X_{k-1}} \quad (5.44)$$

Où : A_k est la matrice de transition d'état.

L'état prédit s'écrit :

$$X_k^- = A_k X_{k-1} \quad (5.45)$$

Une relation entre le vrai état (X_k) et sa prédiction (X_k^-) peut s'écrire comme :

$$X_k^- = X_k - \delta_k \quad (5.46)$$

Avec: δ_k l'erreur entre le vrai état et sa prédiction

La fonction de mesure donnée par l'équation (5.43) est linéarisée en utilisant un développement en série de Taylor du premier ordre autour du vecteur d'état prédit X_k^- :

$$Z_k = h(X_k^-) + \left. \frac{\partial h}{\partial X} \right|_{X=X_k^-} (X_k - X_k^-) + e_k \quad (5.47)$$

Posons : $\left. \frac{\partial h}{\partial X} \right|_{X=X_k^-} = H_k$

L'équation (5.47) peut s'écrire :

$$Z_k = h(X_k^-) + H_k (X_k - X_k^-) + e_k \quad (5.48)$$

Ou

$$Z_k - h(X_k^-) + H_k X_k^- = H_k X_k + e_k \quad (5.49)$$

Etape 2 : Reformulation du filtre

Mettre les équations (5.46) et (5.49) ensemble et sous forme matricielle comme :

$$\begin{bmatrix} X_k^- \\ Z_k - h(X_k^-) + H_k X_k^- \end{bmatrix} = \begin{bmatrix} I \\ H_k \end{bmatrix} X_k + \begin{bmatrix} -\delta_k \\ e_k \end{bmatrix} \quad (5.50)$$

Où : I est la matrice d'identité.

Equation (5.50) est exprimée sous une forme compacte :

$$\check{Z}_k = \check{H}_k X_k + \check{e}_k \quad (5.51)$$

La matrice de covariance de \check{e}_k est donnée par :

$$\check{R}_k = \begin{bmatrix} P_k^- & 0 \\ 0 & R_k \end{bmatrix} = L_k L_k^T \quad (5.52)$$

Avec $R_k = E(e_k e_k^T)$ qui est la matrice de covariance de mesure et $P_k^- = E[\delta_k \delta_k^T]$ qui est la matrice de covariance d'erreur prédite. P_k^- est déterminée par :

$$P_k^- = A_k P_{k-1} A_k^T + Q_k \quad (5.53)$$

Avec Q_k est la matrice de covariance du bruit du processus.

Où L_k est calculée par la factorisation de Cholesky.

Multiplier l'Eq. (5.51) de L_k^{-1} et définissant : $Y_k = L_k^{-1} \check{Z}_k$, $G_k = L_k^{-1} \check{H}_k$, $\zeta_k = L_k^{-1} \check{e}_k$

L'équation de régression est écrite comme :

$$Y_k = G_k X_k + \zeta_k \quad (5.54)$$

Etape 3 : Mise à jour

Cette étape permettra de mettre à jour l'état vrai et de vérifier la position exacte en utilisant l'algorithme RAIM. L'état estimé \hat{X}_k dans l'Eq. (5.54) est optimisé par l'algorithme IRWLS. L'état vrai est estimé par :

$$\hat{X}_k = (G_k^T W_k G_k)^{-1} G_k^T W_k Y_k \quad (5.55)$$

W_k est la matrice des poids qui est déterminée par la méthode MM-Estimation.

La matrice de covariance d'erreurs est calculée par :

$$P_k = (I - KH_k) * P_k^- \quad (5.56)$$

Avec K est le gain du filtre qui est déterminé :

$$K = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \quad (5.57)$$

5.3.2. Application de l'algorithme FKE-robuste

Dans cette section, les données Galileo et les données GPS sont utilisées pour estimer la position du récepteur. Pour combiner les deux systèmes, la position résultante du récepteur sera exprimée dans le même système de coordonnées et de temps. Le référentiel de coordonnées pour le GPS est le WGS-84. Galileo utilise une autre réalisation indépendante : GTRF. En fait, la différence entre les deux systèmes de coordonnées est de quelques centimètres. Alors, on considère que les deux sont similaires. Les systèmes de temps sont également différents. Cependant, la différence des temps est inférieure à 50 ns. Alors, on considère aussi que les deux systèmes de temps sont similaires.

Après avoir éliminé les erreurs GNSS : d'horloge de satellite, ionosphériques, troposphériques et multi-trajets, les pseudo-distances corrigés de deux systèmes (GPS et Galileo) peut s'écrire :

$$\rho_c^{G,E} = r^{G,E} + c\delta t_{G,E} + \varepsilon_{\rho,G,E} \quad (5.58)$$

Où : les exposants et indices G, E se réfèrent aux satellites GPS et Galileo, respectivement ; ρ_c est la mesure de pseudo-distance corrigée ; c est la vitesse de lumière ; δt est le décalage d'horloge du récepteur ; ε_ρ est le bruit de mesure ; $r^{G,E}$ est la vraie pseudo-distance de satellite GPS, Galileo au récepteur :

$$r^{G,E} = \sqrt{(x_{G,E} - x_u)^2 + (y_{G,E} - y_u)^2 + (z_{G,E} - z_u)^2} \quad (5.59)$$

Où $[x_{G,E}, y_{G,E}, z_{G,E}]$ sont les positions des satellites GPS et Galileo ; $[x_u, y_u, z_u]$ constitue la position d'utilisateur.

Etant donné l'effet Doppler, le taux de variation de pseudo-distance $\dot{\rho}$ peut être calculé comme dans [20]:

$$\dot{\rho} = -\frac{Dc}{f} \quad (5.60)$$

Où : D est l'effet Doppler ; c est la vitesse de lumière ; f est la fréquence de fonctionnement du satellite.

La vitesse peut être estimée à partir du taux de variation de pseudo-distance, en commençant par différencier l'équation (5.58) pour obtenir :

$$\dot{\rho}_c^{G,E} = \dot{r}^{G,E} + c\delta\dot{t}_{G,E} + \varepsilon_{\rho,G,E} \quad (5.61)$$

Où : $\delta\dot{t}$ est la dérive d'horloge de l'utilisateur (sec/sec) ; ε_{ρ} est l'erreur d'observation (mètres/sec); \dot{r} est le véritable taux de variation de distance (range rate) exprimé comme suit :

$$\dot{r} = 1_x(v_{x,G,E} - v_{x,u}) + 1_y(v_{y,G,E} - v_{y,u}) + 1_z(v_{z,G,E} - v_{z,u}) \quad (5.62)$$

Où : $[v_{x,G,E}, v_{y,G,E}, v_{z,G,E}]$ est la vitesse du satellite ; $[v_{x,u}, v_{y,u}, v_{z,u}]$ est la vitesse réelle de l'utilisateur ; $[1_x, 1_y, 1_z]$ est le vrai vecteur unité suivant la ligne de visée du satellite vers l'utilisateur :

$$[1_x, 1_y, 1_z] = \frac{[(x_{G,E} - x_u), (y_{G,E} - y_u), (z_{G,E} - z_u)]}{\sqrt{(x_{G,E} - x_u)^2 + (y_{G,E} - y_u)^2 + (z_{G,E} - z_u)^2}} \quad (5.63)$$

Ainsi, les équations combinées pour les pseudo-distances corrigées (Eq.(5.58)) et les taux de variation de pseudo-distances (Eq.(5.61)) donne :

$$\rho_c^{G,E} = r^{G,E} + c\delta t_{G,E} + \varepsilon_{\rho,G,E} \quad (5.64)$$

$$\dot{\rho}_c^{G,E} = \dot{r}^{G,E} + c\delta\dot{t}_{G,E} + \varepsilon_{\dot{\rho},G,E}$$

Pour déterminer la position et la vitesse de l'utilisateur, le vecteur d'état d'utilisateur des deux systèmes peut être défini comme suit :

$$X = [x_u \ v_x \ y_u \ v_y \ z_u \ v_z \ b_G \ d_G \ b_E \ d_E]^T \quad (5.65)$$

Où: $[x_u, y_u, z_u]$ est la position de l'utilisateur; $[v_x, v_y, v_z]$ est la vitesse de l'utilisateur; $b_G = c\delta t_G$ et $b_E = c\delta t_E$ sont les erreurs de distance dues au biais d'horloge de l'utilisateur utilisant respectivement l'horloge GPS et Galileo, et $d_G = c\delta\dot{t}_G$ et $d_E = c\delta\dot{t}_E$ sont la dérive des horloges de l'utilisateur utilisant respectivement l'horloge GPS et Galileo.

Pour appliquer le FKE-robuste à un utilisateur GNSS, nous avons à suivre trois étapes : Linéarisation des équations, reformulation du filtre et mise à jour.

Étape 1 : Linéarisation des équations

On suppose qu'il n'y a pas de couplage entre x_u, y_u, z_u, b_G et b_E . La fonction de transition d'état à l'instant t_k est définie par :

$$X_k^- = f(X_{k-1}) = \begin{bmatrix} x_{u,k-1} + \Delta T * v_{x,k-1} \\ v_{x,k-1} \\ y_{u,k-1} + \Delta T * v_{y,k-1} \\ v_{y,k-1} \\ z_{u,k-1} + \Delta T * v_{z,k-1} \\ v_{z,k-1} \\ b_{G,k-1} + \Delta T * d_{G,k-1} \\ d_{G,k-1} \\ b_{E,k-1} + \Delta T * d_{E,k-1} \\ d_{E,k-1} \end{bmatrix} \quad (5.66)$$

En linéarisant la fonction $f(X_{k-1})$, la matrice de transition d'état A_k est obtenue sous forme d'une matrice diagonale par blocs de 5x5 de la forme :

$$A_k = \left. \frac{\partial f}{\partial X} \right|_{X=X_{k-1}} = \begin{bmatrix} A_x & 0 & 0 & 0 & 0 \\ 0 & A_y & 0 & 0 & 0 \\ 0 & 0 & A_z & 0 & 0 \\ 0 & 0 & 0 & A_{bG} & 0 \\ 0 & 0 & 0 & 0 & A_{bE} \end{bmatrix} \quad (5.67)$$

Où $A_x = A_y = A_z = A_{bG} = A_{bE} = \begin{bmatrix} 1 & \Delta T \\ 0 & 1 \end{bmatrix}$ avec ΔT est l'intervalle de temps entre deux époques.

L'état prédit peut s'écrire :

$$X_k^- = A_k X_{k-1} \quad (5.68)$$

Une relation entre le vrai état (X_k) et sa prédiction (X_k^-) peut s'écrire comme :

$$X_k^- = X_k - \delta_k \quad (5.69)$$

Avec : δ_k l'erreur entre le vrai état et sa prédiction

L'équation d'observation (5.43) est définie comme : $Z_k = h(X_{k-1}) + e_k$

Où

$$Z_k = [Z_{\rho,G} \ Z_{\dot{\rho},G} \ Z_{\rho,E} \ Z_{\dot{\rho},E}]^T \quad (5.70)$$

Avec : $Z_{\rho,G}$ et $Z_{\dot{\rho},G}$ sont les matrices des mesures de pseudo-distances corrigées ($\rho_{c,G}$) et de taux de variation de pseudo-distances ($\dot{\rho}_{c,G}$) de N_{gps} satellites GPS suivis; $Z_{\rho,E}$ et $Z_{\dot{\rho},E}$ sont les matrices des mesures de pseudo-distances corrigées ($\rho_{c,E}$) et de taux de variation de pseudo-distances ($\dot{\rho}_{c,E}$) de N_{gal} satellites Galileo suivis à l'instant t_k .

La fonction de mesure, donnée par l'équation (5.43), est linéarisée en utilisant un développement en série de Taylor du premier ordre autour du vecteur d'état prédit X_k^- (voire la section 5.3.1) :

$$Z_k - h(X_k^-) + H_k X_k^- = H_k X_k + e_k \quad (5.71)$$

Où la matrice H_k est déterminée comme la dérivée de la fonction $h(X_k^-)$: $H_k = \left. \frac{\partial h}{\partial X} \right|_{X=X_k^-}$ peut s'écrire comme :

$$H_k = [H_{k,G} \ H_{k,E}]^T \quad (5.72)$$

Où : $H_{k,G}$ et $H_{k,E}$ sont les Jacobiens du modèle de mesure avec les données GPS et les données Galileo, respectivement, qui sont données par les matrices suivantes :

$$H_{k,G} = \begin{bmatrix} \frac{\partial \rho_G^1}{\partial x} & 0 & \frac{\partial \rho_G^1}{\partial y} & 0 & \frac{\partial \rho_G^1}{\partial z} & 0 & 1 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial \rho_G^{N_{gps}}}{\partial x} & 0 & \frac{\partial \rho_G^{N_{gps}}}{\partial y} & 0 & \frac{\partial \rho_G^{N_{gps}}}{\partial z} & 0 & 1 & 0 & 0 & 0 \\ 0 & \frac{\partial \rho_G^1}{\partial x} & 0 & \frac{\partial \rho_G^1}{\partial y} & 0 & \frac{\partial \rho_G^1}{\partial z} & 0 & 1 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \frac{\partial \rho_G^{N_{gps}}}{\partial x} & 0 & \frac{\partial \rho_G^{N_{gps}}}{\partial y} & 0 & \frac{\partial \rho_G^{N_{gps}}}{\partial z} & 0 & 1 & 0 & 0 \end{bmatrix} \quad (5.73)$$

$$H_{k,E} = \begin{bmatrix} \frac{\partial \rho_E^1}{\partial x} & 0 & \frac{\partial \rho_E^1}{\partial y} & 0 & \frac{\partial \rho_E^1}{\partial z} & 0 & 1 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial \rho_E^{N_{gal}}}{\partial x} & 0 & \frac{\partial \rho_E^{N_{gal}}}{\partial y} & 0 & \frac{\partial \rho_E^{N_{gal}}}{\partial z} & 0 & 1 & 0 & 0 & 0 \\ 0 & \frac{\partial \rho_E^1}{\partial x} & 0 & \frac{\partial \rho_E^1}{\partial y} & 0 & \frac{\partial \rho_E^1}{\partial z} & 0 & 1 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \frac{\partial \rho_E^{N_{gal}}}{\partial x} & 0 & \frac{\partial \rho_E^{N_{gal}}}{\partial y} & 0 & \frac{\partial \rho_E^{N_{gal}}}{\partial z} & 0 & 1 & 0 & 0 \end{bmatrix} \quad (5.74)$$

Où : N_{gps} et N_{gal} sont respectivement le nombre de satellites GPS et Galileo suivis ; les dérivées de pseudo-distance sont :

$$\begin{aligned} \frac{\partial \rho^i}{\partial x} &= \frac{-(x_i - \hat{x}_u^-)}{\sqrt{(x_i - \hat{x}_u^-)^2 + (y_i - \hat{y}_u^-)^2 + (z_i - \hat{z}_u^-)^2}} \\ \frac{\partial \rho^i}{\partial y} &= \frac{-(y_i - \hat{y}_u^-)}{\sqrt{(x_i - \hat{x}_u^-)^2 + (y_i - \hat{y}_u^-)^2 + (z_i - \hat{z}_u^-)^2}} \\ \frac{\partial \rho^i}{\partial z} &= \frac{-(z_i - \hat{z}_u^-)}{\sqrt{(x_i - \hat{x}_u^-)^2 + (y_i - \hat{y}_u^-)^2 + (z_i - \hat{z}_u^-)^2}} \end{aligned} \quad (5.75)$$

$[x_i, y_i, z_i]$ est la position du $i^{\text{ème}}$ satellite (satellites GPS ou Galileo) et $[\hat{x}_u^-, \hat{y}_u^-, \hat{z}_u^-]$ est l'estimation de la position du récepteur à l'instant t_k .

Etape 2 : Reformulation du filtre

En mettant l'équation (5.69) et l'équation (5.71) ensemble, on peut les écrire sous une forme matricielle donnée par :

$$\begin{bmatrix} X_k^- \\ Z_k - h(X_k^-) + H_k X_k^- \end{bmatrix} = \begin{bmatrix} I \\ H_k \end{bmatrix} X_k + \begin{bmatrix} -\delta_k \\ e_k \end{bmatrix} \quad (5.76)$$

Où : I est la matrice d'identité.

L'équation (5.76) est exprimée sous une forme compacte :

$$\check{Z}_k = \check{H}_k X_k + \check{e}_k \quad (5.77)$$

La matrice de covariance de \check{e}_k est donnée par:

$$\check{R}_k = \begin{bmatrix} P_k^- & 0 \\ 0 & R_k \end{bmatrix} = L_k L_k^T \quad (5.78)$$

Où : $R_k = E(e_k e_k^T)$ est la matrice de covariance de mesure et $P_k^- = E[\delta_k \delta_k^T]$ est la matrice de covariance de l'erreur prédite. Pour calculer L_k , on doit déterminer la matrice R_k et P_k^- .

La détermination de R_k n'est pas une tâche facile. Dans ce mémoire, on propose d'utiliser un modèle « Exponentiel », ainsi il est possible d'écrire :

$$S_i = a_1 + a_2 e^{-EL(i)/a_3} \quad (5.79)$$

Où : i est le $i^{\text{ème}}$ satellite ; S_i est la matrice de covariance [m] ; a_1 et a_2 sont des paramètres de modèles [m] ; a_3 est une valeur mise à l'échelle de l'erreur d'élévation [deg] et $EL(i)$ est l'angle d'élévation du $i^{\text{ème}}$ satellite [deg]. a_1 , a_2 , et a_3 doivent être estimés de manière empirique. σ_i^2 est déterminé par :

$$\sigma_i^2 = S_i / \left(\prod_{i=1}^{N_s} S_i \right)^{N_s} \quad (5.80)$$

Où : N_s est le nombre de satellites suivis.

La covariance du bruit de mesure des satellites GPS est déterminée comme suit :

$$R_{k,G} = \begin{bmatrix} \sigma_1^2 & 0 & 0 & 0 \\ 0 & \sigma_2^2 & 0 & 0 \\ 0 & 0 & \vdots & 0 \\ 0 & 0 & 0 & \sigma_{N_{gps}}^2 \end{bmatrix} \quad (5.81)$$

Où : N_{gps} est le nombre de satellites GPS suivis

La covariance du bruit de mesure des satellites Galileo ($R_{k,E}$) est déterminée de la même manière que les satellites GPS. Donc, la matrice de covariance de mesure R_k est :

$$R_k = \begin{bmatrix} R_{k,G} & 0 & 0 & 0 \\ 0 & R_{k,G} & 0 & 0 \\ 0 & 0 & R_{k,E} & 0 \\ 0 & 0 & 0 & R_{k,E} \end{bmatrix} \quad (5.82)$$

La prédiction de covariance d'erreur P_k^- est déterminée :

$$P_k^- = A_k P_{k-1} A_k^T + Q_k \quad (5.83)$$

Q_k est la matrice de covariance des processus.

$$Q_k = \int_{t_{k-1}}^{t_k} A(\tau) \sum A^T(\tau) d\tau \quad (5.84)$$

La matrice Σ est la matrice de densité spectrale [41]

$$\Sigma = \text{diagonal}(S_x, S_{vx}, S_y, S_{vy}, S_z, S_{vz}, S_{b,G}, S_{d,G}, S_{b,E}, S_{d,E}) \quad (5.85)$$

Où : $S_x, S_y, S_z, S_{vx}, S_{vy}, S_{vz}$ sont, respectivement, les densités spectrales de puissance du bruit de position et de vitesse dans les coordonnées X, Y, Z; S_b est la densité spectrale la puissance du bruit de biais de l'horloge et S_d est la densité spectrale de puissance du bruit de dérive de fréquence.

La matrice de covariance Q_k donnée par l'équation (5.84) devient :

$$Q_k = \text{diagonal}(Q_x, Q_y, Q_z, Q_{b,G}, Q_{b,E}) \quad (5.86)$$

Où :

$$Q_x = \begin{bmatrix} S_x \Delta T + S_{vx} (\Delta T)^3 / 3 & S_{vx} (\Delta T)^2 / 2 \\ S_{vx} (\Delta T)^2 / 2 & S_{vx} \Delta T \end{bmatrix} \quad (5.87)$$

Q_y et Q_z sont déterminées de la même manière que Q_x ;

$$Q_{b,G} = \begin{bmatrix} S_{b,G} \Delta T + S_{d,G} (\Delta T)^3 / 3 & S_{d,G} (\Delta T)^2 / 2 \\ S_{d,G} (\Delta T)^2 / 2 & S_{d,G} \Delta T \end{bmatrix} \quad (5.88)$$

$Q_{b,E}$ est déterminée de la même manière que $Q_{b,G}$, avec ΔT l'intervalle de temps entre deux époques.

Après le calcul de la matrice de covariance R_k et la prédiction de covariance P_k^- , il est facile de déterminer la matrice L_k dans l'équation (5.78) en utilisant la factorisation de Cholesky.

En multipliant l'Eq. (5.77) par L_k^{-1} , on obtient : $Y_k = L_k^{-1} \check{Z}_k$, $G_k = L_k^{-1} \check{H}_k$, $\zeta_k = L_k^{-1} \check{e}_k$

L'équation de régression est écrite comme :

$$Y_k = G_k X_k + \zeta_k \quad (5.89)$$

Etape 3 : Mise à jour

L'état d'estimation \hat{X}_k dans l'équation (5.89) est optimisé par l'algorithme IRWLS dans la section 5.3.3 :

$$\hat{X}_k = (G_k^T W_k G_k)^{-1} G_k^T W_k Y_k \quad (5.90)$$

W_k est la matrice des poids.

La matrice de covariance d'erreurs est calculée :

$$P_k = (I - K H_k) * P_k^- \quad (5.91)$$

Avec K est le gain du filtre qui est déterminé par :

$$K = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \quad (5.92)$$

5.3.3. L'algorithme IRWLS

Un algorithme des moindres carrés itératifs repondérés (IRWLS) est appliqué pour éliminer les valeurs aberrantes. Un aperçu de la procédure d'estimation robuste est définie en suivant les étapes suivantes [48] :

- 1) Trouvez une estimation initiale X_0 :

$$X_0 = (G_0^T G_0)^{-1} G_0^T Y \quad (5.93)$$

- 2) Les résidus initiaux sont estimés :

$$r_0 = Y - G_0^T X_0 \quad (5.94)$$

- 3) Plusieurs fonctions d'estimation d'échelle sont définies, mais la fonction médiane suivante est utilisée pour calculer la valeur d'échelle initiale ([48], [55], [56]):

$$s_0 = 1.4826 * median(|r_0|) \quad (5.95)$$

- 4) Estimer la matrice de poids initiale en utilisant la fonction de Huber :

$$W_0(i, i) = \begin{cases} 1 & \text{avec } \left| \frac{r_0(i)}{s_0} \right| \leq \alpha \\ \alpha / \left| \frac{r_0(i)}{s_0} \right| & \text{avec } \left| \frac{r_0(i)}{s_0} \right| > \alpha \end{cases} \quad (5.96)$$

Où $i = 1, \dots, n$; n est le nombre d'observations.

- 5) Boucle (j est la $j^{\text{ième}}$ itération)

- a) Mettre à jour la valeur de la matrice G_j avec X_{j-1} ,
- b) Résoudre l'équation en estimant l'inconnue X_j par la méthode des moindres carrés pondérés

$$X_j = (G_j^T W_{j-1} G_j)^{-1} G_j^T W_{j-1} Y \quad (5.97)$$

- c) Calculer le HPL
- d) Si $\|X_j - X_{j-1}\| \leq 0.001$; break
- e) Si $\|X_j - X_{j-1}\| > 0.001$; continue
- f) Mettre à jour les résidus estimés

$$r_j = Y - G_j^T X_j \quad (5.98)$$

- g) Calculer la valeur de l'échelle

$$s_j = 1.4826 * median(|r_j|) \quad (5.99)$$

- h) Mettre à jour la matrice pondérée en utilisant la fonction de Huber

$$W_j(i, i) = \begin{cases} 1 & \text{avec } \left| \frac{r_j(i)}{s_j} \right| \leq \alpha \\ \alpha / \left| \frac{r_j(i)}{s_j} \right| & \text{avec } \left| \frac{r_j(i)}{s_j} \right| > \alpha \end{cases} \quad (5.100)$$

- i) Aller à l'étape a
- 6) Fini
- 7) Si $HPL \leq HAL$ alors, les positions estimées sont acceptées,
Si non, elles sont rejetées.

5.3.4. Résultats de simulation

Pour illustrer l'algorithme, on utilise les données Galileo et les données GPS de la station ABMF respectivement au format RINEX 3.02 et 2.11. Cette station de référence est située en Guadeloupe. Les données RINEX et les éphémérides de diffusion utilisées pour le calcul ont été téléchargées du réseau RGP et CDDIS pour la journée du 1^{er} janvier 2019.

Pour tester la performance de l'algorithme retenu et présenté dans la section précédente, deux scénarios sont adoptés sous l'environnement Matlab :

Scénario #1 : Solution de navigation basée sur les données Galileo avec la méthode des « moindres carrées »,

Scénario #2 : Solution de navigation basée sur les données Galileo et GPS avec la méthode « FKE-robuste ».

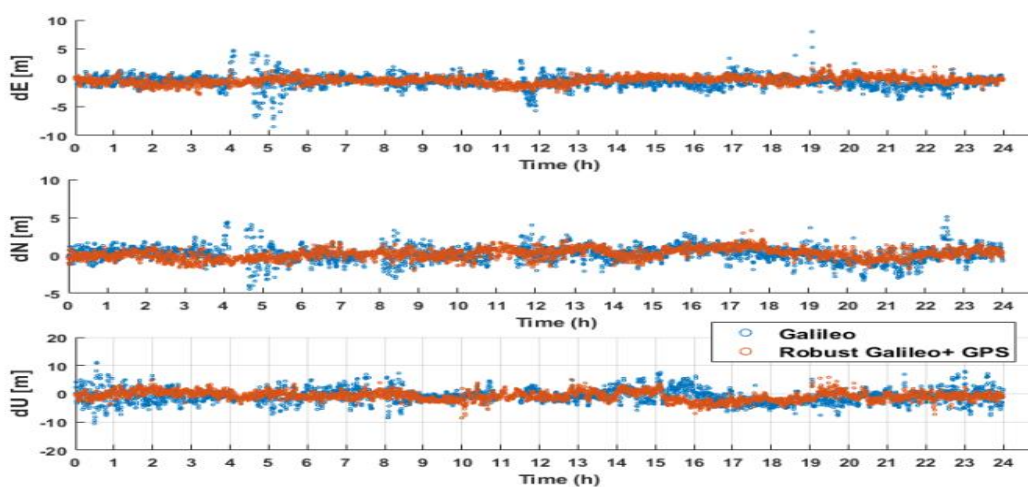


Figure 5.2 : Comparaison des erreurs dans les estimations de position

La Figure 5.2 montre les variations de l'erreur de position du récepteur dans les deux scénarios considérés. La courbe en bleu représente le scénario #1 avec des données Galileo uniquement. On peut voir que les résultats sont instables. Surtout, aux alentours de 4h et 11h, les données Galileo ne peuvent pas déterminer la position ou les erreurs de position sont larges. La combinaison de données Galileo et GPS robustes permet de résoudre ce problème. Les erreurs sont représentées par la courbe orange pour le scénario #2. Un résumé des résultats est donné dans le Tableau 5.4 pour les deux scénarios.

Tableau 5.4 : Erreur RMS en position

Erreur RMS (m)	Galileo	Robust Galileo + GPS
RMS_E	1.3	0.8
RMS_N	1.0	0.7
RMS_U	2.6	1.8
3D-RMS	3.1	2.1

Tableau 5.5 : Erreur Max, Min et l'écart-type

Erreur (m)	Scénario #	
	1	2
Max_E	8.5	2.9
Min_E	9.4e-05	3.9e-05
Ecart type E	0.9	0.5
Max_N	5.1	3.3
Min_N	2.9e-04	2.6e-04
Ecart type N	0.7	0.4
Max_U	11.1	8.6
Min_U	1.1e-04	1.0e-05
Ecart type U	1.6	1.1

Selon les valeurs des erreurs moyenne quadratiques (Erreurs RMS) répertoriées dans le Tableau 5.4, les données Galileo et GPS robustes améliorent la précision de position d'environ 31.3% par rapport à l'utilisation unique des données Galileo.

5.4. Combinaison des algorithmes FKE-robuste et LSTM (Long Short-Term Memory)

Dans la section 5.2, nous avons présenté la combinaison des données GPS et GLONASS avec la méthode des « moindres carrés itérativement repondérés » et dans la section 5.3, nous avons présenté la combinaison des données GPS et Galileo avec la méthode de « filtre de Kalman étendu robuste ». Dans cette section, nous allons compléter le travail en considérant la combinaison des données GPS, Galileo et GLONASS. D'abord, nous effectuerons la comparaison des résultats obtenus en utilisant les deux méthode « FKE-robuste » et les « moindres carrés pondérés ». Ensuite, nous combinerons la méthode « FKE-robuste » et la méthode d'apprentissage profond LSTM (Long Short Term Memory) pour améliorer la précision de position.

5.4.1. Application de la méthode FKE-robuste pour les données issues des trois satellites (GPS, Galileo, GLONASS)

5.4.1.1. Application de la méthode FKE-robuste

De la même manière que dans la section 5.3.2 (combinaison des données GPS et Galileo) et aussi dans la section 5.2 (la synchronisation de la référence de temps GPS et GLONASS), nous avons des équations combinées pour les pseudo-distances corrigées et des taux de pseudo-distances des trois systèmes (GPS, Galileo et GLONASS) :

$$\begin{aligned}\rho_c^{G,E,R} &= r^{G,E,R} + b_{G,E,R} + \varepsilon_{\rho,G,E,R} \\ \dot{\rho}_c^{G,E,R} &= \dot{r}^{G,E,R} + d_{G,E,R} + \varepsilon_{\dot{\rho},G,E,R}\end{aligned}\quad (5.101)$$

Où les exposants et indices G, E, R se réfèrent respectivement aux satellites GPS, Galileo, et GLONASS ; ρ_c et $\dot{\rho}_c$ sont respectivement la mesure de pseudo-distance corrigée et le taux de variation de pseudo-distance ; c est la vitesse de la lumière ; b et d sont respectivement l'erreur de distance dues au biais d'horloge et la dérive des horloges de l'utilisateur ; ε_{ρ} et $\varepsilon_{\dot{\rho}}$ sont respectivement le bruit de mesure et l'erreur d'observation ; $r^{G,E,R}$ et $\dot{r}^{G,E,R}$ sont respectivement la vraie distance et le véritable taux de variation des distances des satellites GPS, Galileo au récepteur.

$$\begin{aligned}r^{G,E,R} &= \sqrt{(x_{G,E,R} - x_u)^2 + (y_{G,E,R} - y_u)^2 + (z_{G,E,R} - z_u)^2} \\ \dot{r}^{G,E,R} &= 1_x(v_{x,G,E,R} - v_{x,u}) + 1_y(v_{y,G,E,R} - v_{y,u}) + 1_z(v_{z,G,E,R} - v_{z,u})\end{aligned}\quad (5.102)$$

Où $[x_{G,E,R}, y_{G,E,R}, z_{G,E,R}]$ sont les positions des satellites GPS, Galileo et GLONASS ; $[x_u, y_u, z_u]$ est la position d'utilisateur ; $[v_{x,G,E,R}, v_{y,G,E,R}, v_{z,G,E,R}]$ est la vitesse du satellite ; $[v_{x,u}, v_{y,u}, v_{z,u}]$ est la vitesse réelle de l'utilisateur ; $[1_x, 1_y, 1_z]$ est le vrai vecteur d'unité de ligne de visée du satellite à l'utilisateur.

De l'équation (5.101), l'état utilisateur de trois systèmes peut être défini comme suit :

$$X = [x_u \ v_x \ y_u \ v_y \ z_u \ v_z \ b_G \ d_G \ b_E \ d_E \ b_R \ d_R]^T \quad (5.103)$$

Pour appliquer la méthode FKE-robuste à un utilisateur GNSS, nous avons à suivre trois étapes : Linéarisons des équations, reformulation du filtre et mise à jour.

Etape 1 : Linéarisation des équations

En linéarisant la fonction $f(X_{k-1})$ de l'équation (5.42), la matrice de transition d'état A_k est obtenue sous forme d'une matrice diagonale par blocs de 6x6 de la forme :

$$A_k = \left. \frac{\partial f}{\partial X} \right|_{X=X_{k-1}} = \begin{bmatrix} A_x & 0 & 0 & 0 & 0 & 0 \\ 0 & A_y & 0 & 0 & 0 & 0 \\ 0 & 0 & A_z & 0 & 0 & 0 \\ 0 & 0 & 0 & A_{b,G} & 0 & 0 \\ 0 & 0 & 0 & 0 & A_{b,E} & 0 \\ 0 & 0 & 0 & 0 & 0 & A_{b,R} \end{bmatrix} \quad (5.104)$$

Où $A_x = A_y = A_z = A_{b,G} = A_{b,E} = A_{b,R} = \begin{bmatrix} 1 & \Delta T \\ 0 & 1 \end{bmatrix}$ avec ΔT est l'intervalle de temps entre deux époques.

L'état prédit peut s'écrire :

$$X_k^- = A_k X_{k-1} \quad (5.105)$$

Une relation entre le vrai état (X_k) et sa prédiction (X_k^-) peut s'écrire comme :

$$X_k^- = X_k - \delta_k \quad (5.106)$$

Avec: δ_k l'erreur entre le vrai état et sa prédiction.

L'équation d'observation (5.43) est définie comme : $Z_k = h(X_{k-1}) + e_k$

Où

$$Z_k = [Z_{\rho,G} \ Z_{\dot{\rho},G} \ Z_{\rho,E} \ Z_{\dot{\rho},E} \ Z_{\rho,R} \ Z_{\dot{\rho},R}]^T \quad (5.107)$$

Avec : $Z_{\rho,G}$ et $Z_{\dot{\rho},G}$ sont les matrices des mesures de pseudo-distances corrigées ($\rho_{c,G}$) et de taux de variation de pseudo-distance ($\dot{\rho}_{c,G}$) de N_{gps} satellites GPS suivis; $Z_{\rho,E}$ et $Z_{\dot{\rho},E}$ sont les matrices des mesures de pseudo-distances corrigées ($\rho_{c,E}$) et de taux de variation de pseudo-distance ($\dot{\rho}_{c,E}$) de N_{gal} satellites Galileo suivis ; $Z_{\rho,R}$ et $Z_{\dot{\rho},R}$ sont les matrices des mesures de pseudo-distances corrigées ($\rho_{c,R}$) et de taux de variation de pseudo-distance ($\dot{\rho}_{c,R}$) de N_{glo} satellites GLONASS suivis à l'instant t_k .

La fonction de mesure donnée par l'équation (5.43) est linéarisée en utilisant un développement en série de Taylor du premier ordre autour du vecteur d'état prédit X_k^- (Cf. la section 5.3.1) :

$$Z_k - h(X_k^-) + H_k X_k^- = H_k X_k + e_k \quad (5.108)$$

Où la matrice H_k est déterminée comme la dérivée de la fonction $h(X_k^-)$: $H_k = \left. \frac{\partial h}{\partial X} \right|_{X=X_k^-}$ peut s'écrire comme :

$$H_k = \begin{bmatrix} H_{\rho,G} & 1 & 0 & 0 & 0 & 0 & 0 \\ H_{\dot{\rho},G} & 0 & 1 & 0 & 0 & 0 & 0 \\ H_{\rho,E} & 0 & 0 & 1 & 0 & 0 & 0 \\ H_{\dot{\rho},E} & 0 & 0 & 0 & 1 & 0 & 0 \\ H_{\rho,R} & 0 & 0 & 0 & 0 & 1 & 0 \\ H_{\dot{\rho},R} & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.109)$$

Où: $H_{\rho,G}$, $H_{\rho,E}$ et $H_{\rho,R}$ sont les Jacobiens du modèle de pseudo-distance avec respectivement les données GPS, Galileo et GLONASS avec les dimensions $(N_{gps} \times 6)$, $(N_{gal} \times 6)$, $(N_{glo} \times 6)$; $H_{\dot{\rho},G}$, $H_{\dot{\rho},E}$ et $H_{\dot{\rho},R}$ sont les Jacobiens du modèle de taux de pseudo-distance avec respectivement les données GPS, Galileo et GLONASS avec les dimensions $(N_{gps} \times 6)$, $(N_{gal} \times 6)$, $(N_{glo} \times 6)$;

$H_{\rho,G}$ est calculée par :

$$H_{\rho,G} = \begin{bmatrix} \frac{\partial \rho_G^1}{\partial x} & 0 & \frac{\partial \rho_G^1}{\partial y} & 0 & \frac{\partial \rho_G^1}{\partial z} & 0 \\ \frac{\partial \rho_G^2}{\partial x} & 0 & \frac{\partial \rho_G^2}{\partial y} & 0 & \frac{\partial \rho_G^2}{\partial z} & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial \rho_G^{N_{gps}}}{\partial x} & 0 & \frac{\partial \rho_G^{N_{gps}}}{\partial y} & 0 & \frac{\partial \rho_G^{N_{gps}}}{\partial z} & 0 \end{bmatrix} \quad (5.110)$$

Et $H_{\dot{\rho},G}$ est calculée par :

$$H_{\dot{\rho},G} = \begin{bmatrix} 0 & \frac{\partial \rho_G^1}{\partial x} & 0 & \frac{\partial \rho_G^1}{\partial y} & 0 & \frac{\partial \rho_G^1}{\partial z} \\ 0 & \frac{\partial \rho_G^2}{\partial x} & 0 & \frac{\partial \rho_G^2}{\partial y} & 0 & \frac{\partial \rho_G^2}{\partial z} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \frac{\partial \rho_G^{N_{gps}}}{\partial x} & 0 & \frac{\partial \rho_G^{N_{gps}}}{\partial y} & 0 & \frac{\partial \rho_G^{N_{gps}}}{\partial z} \end{bmatrix} \quad (5.111)$$

$$\text{Où } \frac{\partial \rho_{c,G}^i}{\partial x} = \frac{-(x_G^i - x_u)}{\sqrt{(x_G^i - x_u)^2 + (y_G^i - y_u)^2 + (z_G^i - z_u)^2}}; \frac{\partial \rho_{c,G}^i}{\partial y} = \frac{-(y_G^i - y_u)}{\sqrt{(x_G^i - x_u)^2 + (y_G^i - y_u)^2 + (z_G^i - z_u)^2}}; \frac{\partial \rho_{c,G}^i}{\partial z} = \frac{-(z_G^i - z_u)}{\sqrt{(x_G^i - x_u)^2 + (y_G^i - y_u)^2 + (z_G^i - z_u)^2}} \text{ avec } i = 1, \dots, N_{gps}.$$

$H_{\rho,E}$, $H_{\rho,R}$ et $H_{\dot{\rho},E}$, $H_{\dot{\rho},R}$ sont calculées respectivement comme pour $H_{\rho,G}$ et $H_{\dot{\rho},G}$.

Etape 2 : Reformulation du filtre

En mettant ensemble les équations (5.128) et (5.130), nous pouvons écrire sous forme matricielle :

$$\begin{bmatrix} X_k^- \\ Z_k - h(X_k^-) + H_k X_k^- \end{bmatrix} = \begin{bmatrix} I \\ H_k \end{bmatrix} X_k + \begin{bmatrix} -\delta_k \\ e_k \end{bmatrix}, \text{ cette équation est alors exprimée sous une forme compacte :}$$

$$\check{Z}_k = \check{H}_k X_k + \check{e}_k \quad (5.112)$$

La matrice de covariance de \check{e}_k est donnée par :

$$\check{R}_k = \begin{bmatrix} P_k^- & 0 \\ 0 & R_k \end{bmatrix} = L_k L_k^T \quad (5.113)$$

Où $R_k = E(e_k e_k^T)$ est la matrice de covariance de mesure et $P_k^- = E[\delta_k \delta_k^T]$ est la matrice de covariance de l'erreur prédite. Pour calculer L_k , on doit déterminer la matrice R_k et P_k^- .

La détermination de R_k n'est pas une tâche facile. Dans ce mémoire on propose un modèle « Exponentiel » écrit comme dans l'équation (5.79) : $S_i = a + b e^{-E l(i)/c}$. σ_i^2 est déterminé via l'équation (5.80) : $\sigma_i^2 = S_i / (\prod_{i=1}^{N_s} S_i)^{N_s}$. La covariance du bruit de mesure des satellites GPS

est déterminée par l'équation (5.81) : $R_{k,G} = \begin{bmatrix} \sigma_1^2 & 0 & 0 & 0 \\ 0 & \sigma_2^2 & 0 & 0 \\ 0 & 0 & \vdots & 0 \\ 0 & 0 & 0 & \sigma_{N_{gps}}^2 \end{bmatrix}$. Les covariances du bruit

des mesures des satellites Galileo ($R_{k,E}$) et GLONASS $R_{k,R}$ sont déterminées de la même manière que pour les satellites GPS. Ainsi, la matrice de covariance de mesure R_k est donnée par :

$$R_k = \begin{bmatrix} R_{k,G} & 0 & 0 & 0 & 0 & 0 \\ 0 & \gamma R_{k,G} & 0 & 0 & 0 & 0 \\ 0 & 0 & R_{k,E} & 0 & 0 & 0 \\ 0 & 0 & 0 & \gamma R_{k,E} & 0 & 0 \\ 0 & 0 & 0 & 0 & R_{k,R} & 0 \\ 0 & 0 & 0 & 0 & 0 & \gamma R_{k,R} \end{bmatrix} \quad (5.114)$$

Où $\gamma R_{k,G}$, $\gamma R_{k,E}$ et $\gamma R_{k,R}$ sont, respectivement, des matrices de taux de variation de pseudo-distance de covariances GPS, Galileo et GLONASS. γ est le taux de variation de pseudo-distance de covariance par rapport à la pseudo-distance de covariance, qui est choisi empiriquement ; après avoir effectué plusieurs tests, la valeur $\gamma = 0.01$ a été choisie dans ce mémoire.

La prédiction de covariance d'erreur P_k^- est déterminée par l'équation (5.83) : $P_k^- = A_k P_{k-1} A_k^T + Q_k$. Q_k est la matrice de covariance des processus, comme dans l'équation (5.84) : $Q_k = \int_{t_{k-1}}^{t_k} A(\tau) \Sigma A^T(\tau) d\tau$. La matrice Σ est la matrice de densité spectrale similaire à l'équation (5.85) :

$$\Sigma = \text{diagonal}(S_x, S_{vx}, S_y, S_{vy}, S_z, S_{vz}, S_{b,G}, S_{d,G}, S_{b,E}, S_{d,E}, S_{b,R}, S_{d,R}) \quad (5.115)$$

La matrice de covariance Q_k devient :

$$Q_k = \text{diagonal}(Q_x, Q_y, Q_z, Q_{b,G}, Q_{b,E}, Q_{b,R}) \quad (5.116)$$

Où Q_x est calculée comme l'équation (5.87) : $Q_x = \begin{bmatrix} S_x \Delta T + S_{vx} (\Delta T)^3 / 3 & S_{vx} (\Delta T)^2 / 2 \\ S_{vx} (\Delta T)^2 / 2 & S_{vx} \Delta T \end{bmatrix}$

Q_y et Q_z sont déterminées de la même manière que pour Q_x ; $Q_{b,G}$ calculées à partir de l'équation (5.88) : $Q_{b,G} = \begin{bmatrix} S_{b,G} \Delta T + S_{d,G} (\Delta T)^3 / 3 & S_{d,G} (\Delta T)^2 / 2 \\ S_{d,G} (\Delta T)^2 / 2 & S_{d,G} \Delta T \end{bmatrix}$ et $Q_{b,E}$ et $Q_{b,R}$ sont déterminées de la même manière que pour $Q_{b,G}$, avec ΔT l'intervalle de temps entre deux époques.

Après le calcul de la matrice de covariance R_k et la prédiction de covariance P_k^- , il est facile de déterminer la matrice L_k dans l'équation (5.135) en utilisant la factorisation de Cholesky.

En multipliant l'Eq. (5.112) par L_k^{-1} , on obtient : $Y_k = L_k^{-1} \check{Z}_k$, $G_k = L_k^{-1} \check{H}_k$, $\zeta_k = L_k^{-1} \check{e}_k$

L'équation de régression est écrite comme :

$$Y_k = G_k X_k + \zeta_k \quad (5.117)$$

Etape 3 : Mise à jour

L'état d'estimation \hat{X}_k est optimisé par l'algorithme IRWLS dans la section 5.3.3 :

$$\hat{X}_k = (G_k^T W_k G_k)^{-1} G_k^T W_k Y_k \quad (5.118)$$

W_k est la matrice des poids qui est présentée dans la section 5.3.3 et nous avons pris la fonction de poids de Huber pour la combinaison des données GPS et Galileo. Mais, dans cette section, on va remplacer la fonction de Huber par Bi-Tukey pour la combinaison des données GPS, Galileo et GLONASS. La matrice de covariance d'erreur est calculée : $P_k = (I - KH_k) * P_k^-$ avec K est le gain du filtre qui est déterminé par : $K = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1}$.

5.4.1.2. Résultats de simulation

Pour illustrer l'algorithme présenté ci-dessus, cette étude utilise les données de GPS, de Galileo et de GLONASS récoltées à partir de la station ABMF (format RINEX version 2.11 (GPS et GLONASS) et version 3.02 (Galileo)). Cette station de référence est située en Guadeloupe. Les données RINEX ont été téléchargées du réseau RGP et du CDDIS pour la journée du 1^{er} janvier 2019. Afin d'évaluer les performances de l'algorithme présenté ci-dessus, cinq scénarios sont considérés et des simulations ont été effectuées sous l'environnement MATLAB :

- Scénario #1 : Solution de navigation basée sur les données GPS,
- Scénario #2 : Solution de navigation basée sur les données Galileo,
- Scénario #3 : Solution de navigation basée sur les données GLONASS,
- Scénario #4 : Solution de navigation basée sur les données GPS/Galileo/GLONASS,
- Scénario #5 : Solution de navigation basée sur les données GPS/Galileo/GLONASS à l'aide de FKE-robuste.

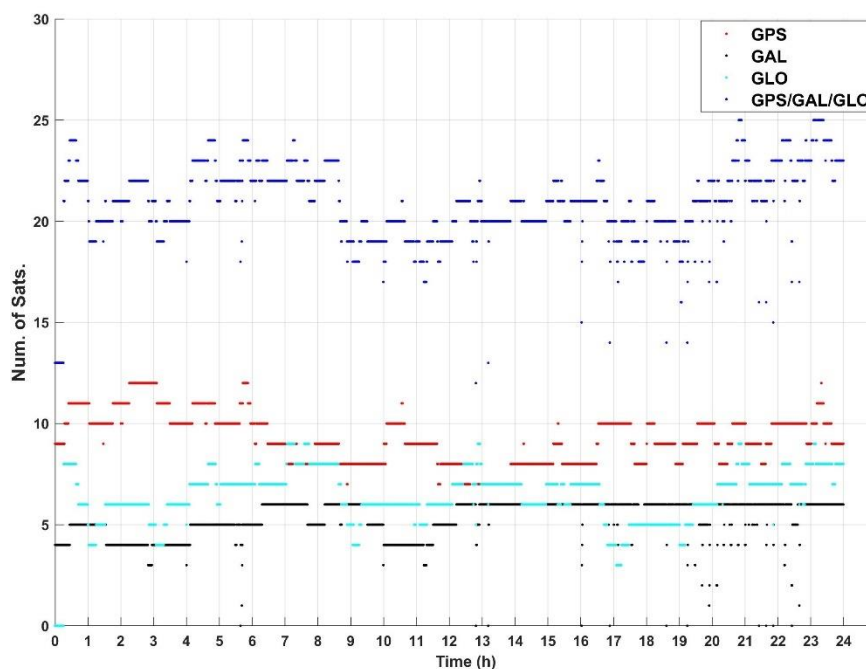


Figure 5.3 : Nombre de satellite visibles

La Figure 5.3 illustre le nombre de satellites visibles du GPS (courbe en rouge), de Galileo (courbe en noir), GLONASS (courbe en cyan), et les systèmes combinés GPS/Galileo/GLONASS (courbe en bleu) sur un masque d'élévation de 10° . Elle montre que le nombre de satellites Galileo et GLONASS est égal à zéro à certaines époques. Le nombre moyen de satellites visibles est d'environ 20 satellites quand on combine les trois systèmes de satellites GPS, Galileo et GLONASS pour déterminer la position et la vitesse du récepteur. Un

grand nombre de satellites présente un avantage puisqu'il apporte plus de signaux vers le récepteur. Cependant, il présente également un inconvénient, car il peut apporter des signaux défaillants ce qui peut pénaliser la précision de l'estimation de la position. Les résultats sont présentés dans la Figure 5.4 qui représente les erreurs de position des cinq scénarios par rapport au temps. dE, dN et dU sont les erreurs à l'Est, au Nord et à la verticale (Up) des estimations de la position du récepteur.

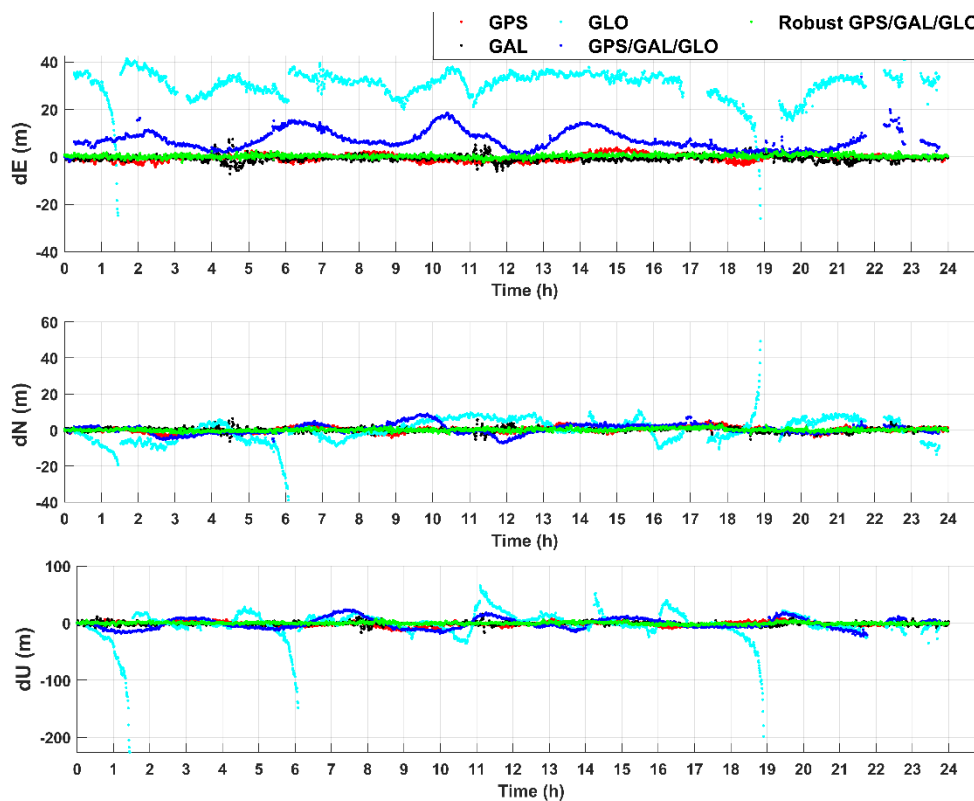


Figure 5.4 : Erreurs de position des cinq scénarios à la station ABMF.

Dans la Figure 5.4, la courbe en rouge se rapporte au scénario #1 avec les données GPS uniquement. La courbe en noir représente le scénario #2 avec des données Galileo uniquement. La courbe en cyan est liée au scénario #3 avec les données GLONASS uniquement. On peut constater que les erreurs de position des données GPS et Galileo sont faibles et stables alors que celles des données GLONASS sont volumineuses et instables. La courbe en bleu représente pour le scénario #4 avec combinaison de données GPS, Galileo et GLONASS. La précision de la position obtenue grâce à cette combinaison s'améliore par rapport à l'utilisation unique des données GLONASS. Cependant, les erreurs restent instables et importantes dans l'axe Est. En particulier, les positions ne sont pas déterminées autour des heures 22h, 23h et 24h, en raison de la plus grande erreur sur le niveau HPL (dans la Figure 5.5). Par conséquent, la méthode FKE-robuste est appliquée pour améliorer la précision de position dans le scénario #5 et les erreurs de position sont représentées dans la courbe en vert. Elles sont plus stables et plus précises que pour les autres scénarios.

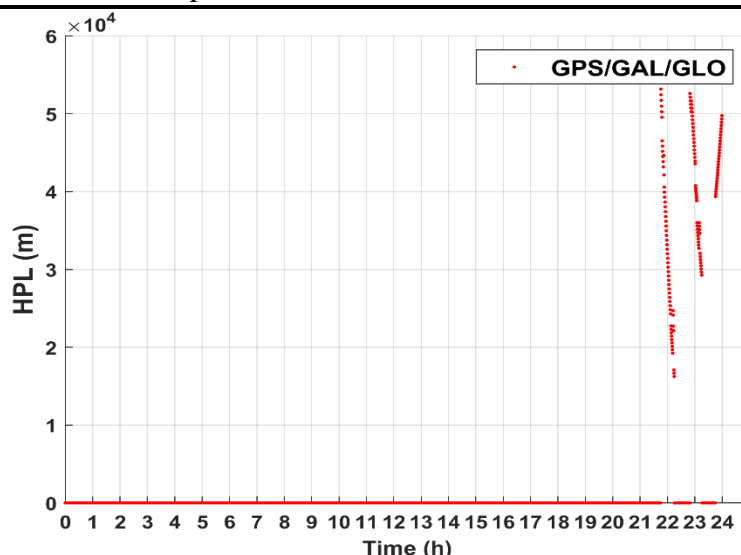


Figure 5.5 : HPLs de données combinées GPS/Galileo/GLONASS à la station ABMF.

La Figure 5.5 représente les HPLs des données combinées de GPS/Galileo/GLONASS en fonction du temps. Les HPLs sont représentés pour vérifier une zone assurée de la position estimée. Si le HPL est plus grand que le HAL, les signaux de défaillance sont détectés et affectent la précision de la position. La valeur de HAL dépend de certaines opérations typiques (Tableau 5.6).

Tableau 5.6 : Exigences de performance du service de sécurité de la vie de EGNOS [58]

Opération typique	HAL (Horizontal Alert Limit)	VAL (vertical Alert limit)
En-route (océance, faible densité continentale)	7.4 km	N/A
En-route (continentale)	3.7 km	N/A
En-route (terminal)	1.85 km	N/A
Approche initiale, approche intermédiaire, approche de non-précision, départ	556m	N/A
Opérations d'approche avec guidage vertical (APV-I)	40 m	50 m
Approche de précision de catégorie I	40 m	35.0 m to 10.0 m

On utilise les données de la station de base, on peut alors mettre $HAL = 20$ (m). Après avoir appliqué la méthode FKE-robuste, les HPLs des données robustes de GPS, Galileo et GLONASS sont recalculés dans la Figure 5.6 et les grands HPLs sont réduits.

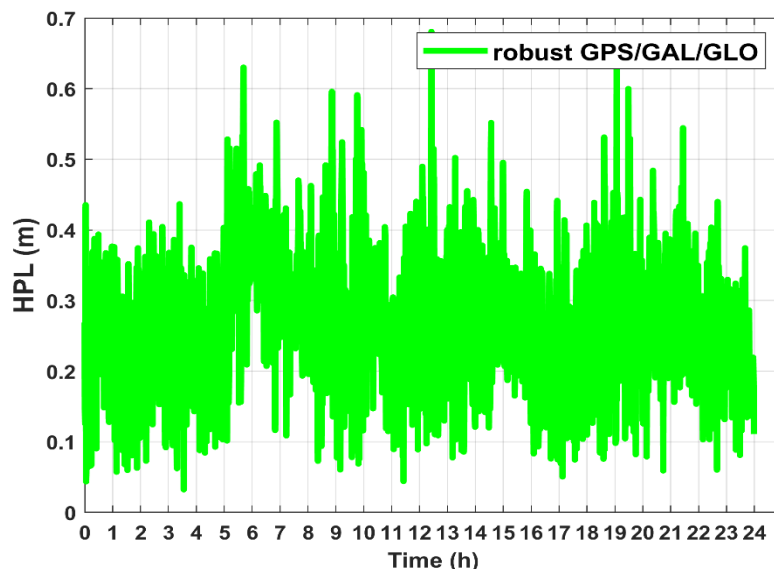


Figure 5.6 : HPLs de données GPS/GAL/GLO robustes à la station ABMF.

Pour plus d'informations sur la précision de la position obtenue pour les cinq scénarios, le Tableau 5.7 présente une comparaison des erreurs de position correspondantes.

Tableau 5.7 : Erreurs de position pour cinq scénarios à la station ABMF en 24h.

Erreurs RMS (m)	GPS	GAL	GLO	GPS/GAL/GLO	Robuste-GPS/GAL/GLO
RMS_E	1.4	1.3	31.0	8.1	0.7
RMS_N	1.5	1.2	6.7	2.7	0.8
RMS_U	3.9	3.1	25.7	9.5	1.8
3D-RMS	4.4	3.6	40.8	12.7	2.1

Tableau 5.8 : Erreur Max, Min et l'écart-type

Erreur (m)	Scénario #				
	1	2	3	4	5
Max_E	4.3	7.4	42.5	33.7	3.3
Min_E	1.4e-04	7.6e-04	0.3	0.03	6.6e-04
Ecart type E	0.8	0.9	5.6	4.1	0.5
Max_N	5.2	6.4	49.2	9.1	3.0
Min_N	2.7e-04	9.7e-04	4.2e-04	1.7e-04	3.5e-07
Ecart type N	0.9	0.8	4.1	1.8	0.5
Max_U	13.8	17.2	226.8	23.5	7.8
Min_U	25e-04	7e-04	15e-03	32e-04	8.2e-05
Ecart type U	2.6	2.0	20.8	4.9	1.1

Selon les valeurs d'erreur RMS (Root Mean Square) indiquées dans le Tableau 5.7, la combinaison robuste des données GPS, Galileo et GLONASS améliore la précision de la position de la position d'environ 53%, 41% et 95%, respectivement par rapport à l'utilisation des données GPS uniquement, des données Galileo uniquement et des données GLONASS uniquement. Elle est également améliorée d'environ 84% par rapport à la combinaison non-robuste de données GPS, Galileo et GLONASS.

Pour confirmer la supériorité de cette méthode, on la compare ci-dessous avec une autre méthode. Dans cette étude, on compare cette méthode avec la méthode « moindres carrés pondérées » (WLSE-Weighted Least-Squares Estimation) et les résultats sont illustrés dans la section 4.3.

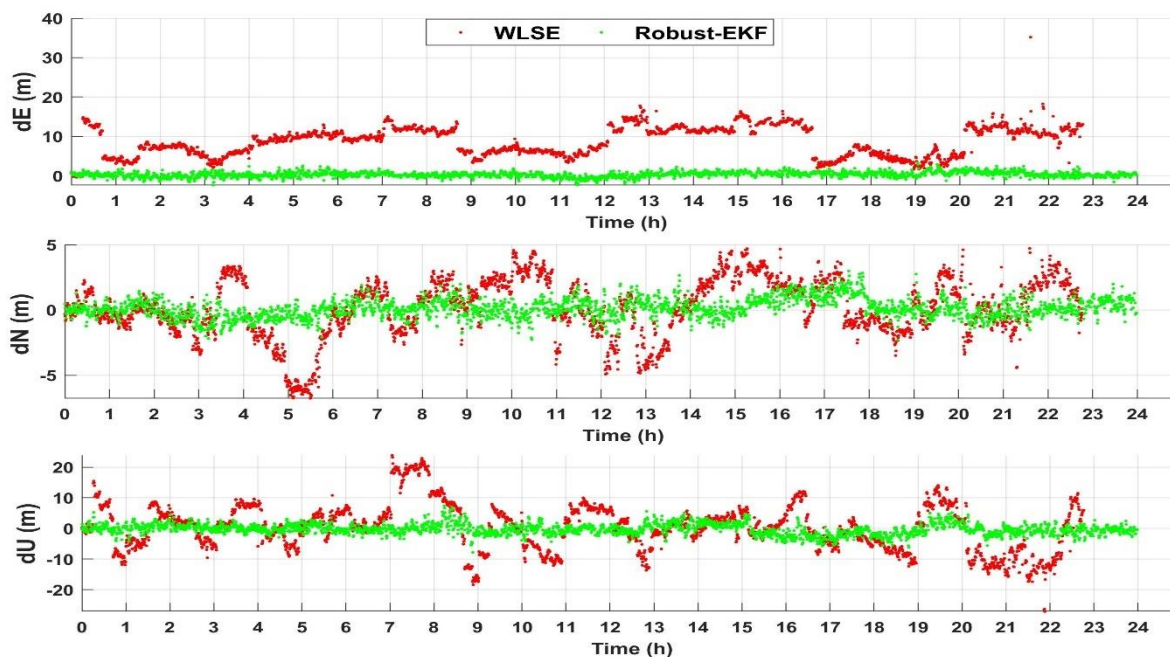


Figure 5.7 : Comparaison des erreurs de position entre FKE-robuste (Robust-EKF) et WLSE.

La Figure 5.7 présente les erreurs de position obtenues à partir des deux méthodes en utilisant les données GPS, Galileo et GLONASS. La méthode WLSE a été développée dans la section 4.3, où la matrice des poids est définie à l'aide du « modèle exponentiel ». Les erreurs de position de WLSE sont fluctuantes ; en particulier, les positions autour de 23h-24h ne sont pas disponibles en raison des grands HPLs. D'autre part, les erreurs de position obtenues avec FKE-robuste sont stables et faibles.

Bien que la FKE-robuste puisse améliorer considérablement la précision de la position, ses résultats pourraient contenir des erreurs qui ne peuvent pas être estimées. Il faut donc un algorithme qui puisse prédéterminer les erreurs de comportement de la méthode FKE-robuste. Pour ce faire, une méthode de dé-bruitage de filtre est présentée dans la section 5.4.2.

5.4.2. Méthode de dé-bruitage de filtre

Les récents développements dans le domaine « l'apprentissage profond » ont permis l'utilisation de ses algorithmes pour les applications GNSS [59][60][61][62]. Dans [62], un système hybride intelligent consistant en un système comprenant le réseau neuronal et le filtre

de Kalman (KF) ont été proposés pour améliorer la précision des composantes de position ainsi que des composantes d'orientation en temps réel. Ce système permet de surmonter les limites du filtre de Kalman. Dans [60], un algorithme « d'apprentissage profond » a été proposé pour débruiter les signaux de sortie des MEMS IMU. L'algorithme utilisé est une méthode « Long Short-term Memory » (LSTM) qui a été employée pour filtrer les sorties des gyroscopes MEMS, dans lesquels les signaux ont été traités comme des séries temporelles. LSTM est une architecture de réseau neuronal récurrent artificiel (RNN) qui est utile pour classer, traiter et faire des prédictions basées sur des données de séries temporelles. Par conséquent, avec une combinaison des travaux présentés dans [60] et [62], LSTM est mise œuvre dans ce mémoire pour apprendre et compenser les erreurs résiduelles de la robustesse de la méthode FKE afin d'améliorer la précision de la position. Ainsi, les schémas hybrides FKE-robuste et LSTM (FKER-LSTM) sont proposées comme une méthode capable d'apprendre comment le vecteur d'état se comporte en fonction de la dynamique du filtre. Les détails des schémas hybrides FKER-LSTM et les résultats expérimentaux pour des données GNSS sont présentés dans les sections suivantes 5.4.2.1 et 5.4.2.2.

5.4.2.1. Modèle de dé-bruitage de filtre

La Figure 5.8 présente l'architecture en couches du LSTM. Cette conception montre le déroulement d'un temps série X (entrées) avec N caractéristiques de longueur S à travers une couche LSTM. Les sorties sont une série chronologique « h » avec H unités cachées de longueur S. dans la conception, h_k et c_k désignent, respectivement la sortie (état caché) et « l'état de la cellule » à l'étape t_k . Le premier bloc LSTM utilise l'état initial du réseau pour calculer la première sortie (h_1) et le premier état de la cellule mis à jour (c_1). Au t_k , le bloc utilise l'état précédent (h_{k-1}, c_{k-1}) pour déterminer la sortie (h_k) et « l'état de la cellule mis à jour » (c_k). Une unité de LSTM est composée d'une « cellule », d'une « porte d'entrée », d'une « porte de sortie » et d'une « porte d'oubli ». La structure de base de l'unité LSTM est présentée par la Figure 5.9, qui illustre le flux de données à l'instant t_k . Ce diagramme met en évidence les trois parties : porte d'entrée, porte de sortie et porte d'oubli.

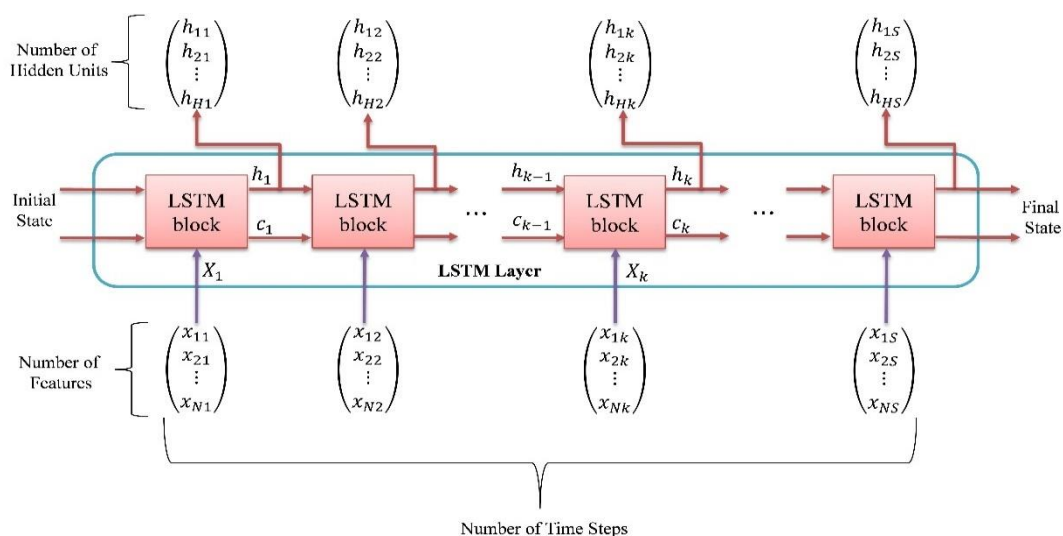


Figure 5.8 : Architecture en couches LSTM (source : mathworks/lstm)

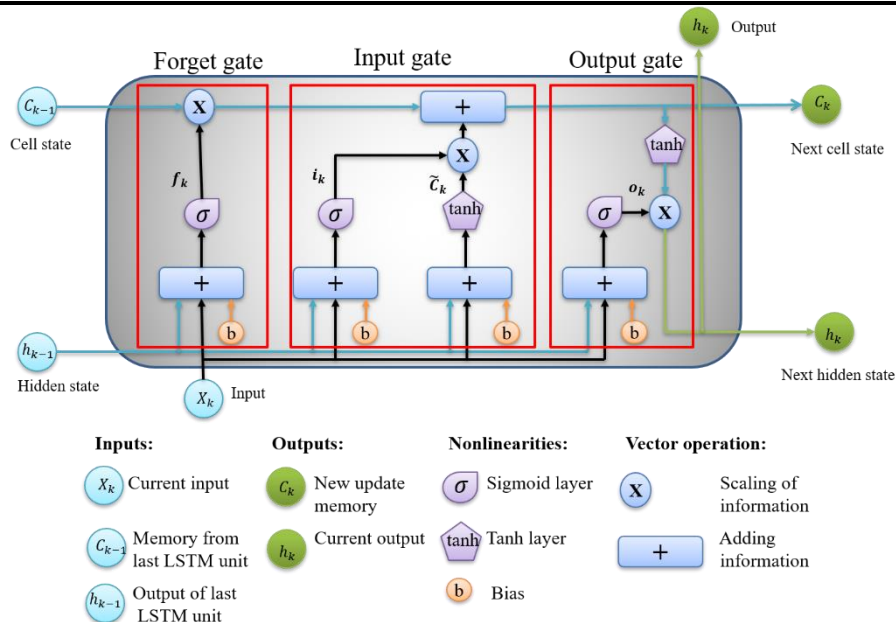


Figure 5.9 : Base de structure d'unité de LSTM

Une couche LSTM possède les poids et les biais appris : les poids d'entrée W_{lstm} , les poids récurrents R_{lstm} et les biais b_{lstm} . Les matrices W_{lstm} , R_{lstm} , et b_{lstm} sont l'union des poids d'entrée, les poids récurrents et les biais de chaque partie, respectivement. Ces matrices sont regroupées comme suit :

$$W = \begin{bmatrix} W_f \\ W_i \\ W_c \\ W_o \end{bmatrix}; R = \begin{bmatrix} R_f \\ R_i \\ R_c \\ R_o \end{bmatrix}; b = \begin{bmatrix} b_f \\ b_i \\ b_c \\ b_o \end{bmatrix} \quad (5.119)$$

Où f , i , c et o désignent respectivement la porte d'oubli (forget), la porte d'entrée (input), l'état de la cellule (cell) et la porte de sortie (output).

Dans chaque bloc de LSTM, l'objectif est le calcul de la sortie (état caché) et de l'état de la cellule à travers trois parties : la porte d'oubli, la porte d'entrée et la porte de sortie. Comme le montre la Figure 5.9, la « porte d'oubli » est présentée dans **la première partie** du LSTM, qui sert à décider quelles informations seront conservées dans l'état de la cellule. La décision est prise par une couche sigmoïde appelée « couche de la porte d'oubli ». h_{k-1} et x_k sont les entrées de la fonction sigmoïde, et la sortie est une valeur allant de 0 à 1 pour chaque nombre dans l'état de la cellule C_{k-1} . Si le résultat est « 1 », l'information est « entièrement conservée » à l'état de cellule. Lorsque la sortie est « 0 », l'information est « complètement effacée ». Le vecteur d'activation de la porte de l'oubli f_k s'écrit :

$$f_k = \sigma(W_f x_k + R_f h_{k-1} + b_f) \quad (5.120)$$

Où : $\sigma(\cdot)$ est la fonction sigmoïde, h_{k-1} est l'état caché au temps t_{k-1} et x_k est le vecteur d'entrée au temps t_k .

La deuxième partie est la « porte d'entrée », qui est utilisée pour décider quelles nouvelles informations doivent entrer de l'état de la cellule précédente. Cette porte est composée de deux parties : (1) une couche « sigmoïde » pour décider les valeurs qui seront mises à jour. Les valeurs de sortie i_k pour cette couche vont de 0 à 1. « 0 » signifie « pas important » et « 1 » représente « important » ; (2) une autre partie est la couche « tanh » qui crée un vecteur de

nouvelles valeurs candidates \tilde{C}_k entre -1 et 1 pour aider à réguler le réseau. Le vecteur d'activation de la porte d'entrée i_k et le nouveau vecteur candidat \tilde{C}_k sont calculés comme suit :

$$i_k = \sigma(W_i x_k + R_i h_{k-1} + b_i) \quad (5.121)$$

$$\tilde{C}_k = \tanh(W_c x_k + R_c h_{k-1} + b_c) \quad (5.122)$$

Où « tanh » est la fonction tangente hyperbolique.

Le nouvel état de cellule C_k est mis à jour par rapport à l'ancien état de cellule C_{k-1} comme suit :

$$C_k = f_k * C_{k-1} + i_k * \tilde{C}_k \quad (5.123)$$

Dans **la troisième et dernière partie**, la « porte de sortie » décidera de la sortie. Tout d'abord, une couche sigmoïde est utilisée pour déterminer quelles parties de l'état de la cellule seront sorties. Ensuite, l'état de la cellule est passé par une fonction « tanh » et la plage des valeurs de l'état de la cellule est de -1 à 1. Enfin, les résultats sont multipliés par la sortie de la porte sigmoïde, et les parties de la sortie sont décidées. Le vecteur d'activation et le vecteur d'état caché de la porte de sortie sont donnés par :

$$o_k = \sigma(W_o x_k + R_o h_{k-1} + b_o) \quad (5.124)$$

$$h_k = o_k * \tanh C_k \quad (5.125)$$

Les équations (5.120)-(5.125), données ci-dessus, constituent les étapes de calcul de la sortie et de l'état des cellules d'une unité de LSTM. Le réseau LSTM peut être appliqué pour des tâches de classification ou de régression avec des séquences et des temps séries des données. Dans ce mémoire, on les utilise pour la tâche de régression. A partir de la Figure 5.8, on obtient les résultats « h » avec les dimensions $(H \times S)$, alors que l'on veut avoir des résultats prédits « Y » avec des dimensions $(M \times S)$; donc on a besoin d'une matrice de poids de dimension $(M \times H)$ pour transformer « h » en « Y ». En conséquence, nous avons l'architecture complète du réseau LSTM avec une couche LSTM pour la régression, comme le montre la Figure 5.10.

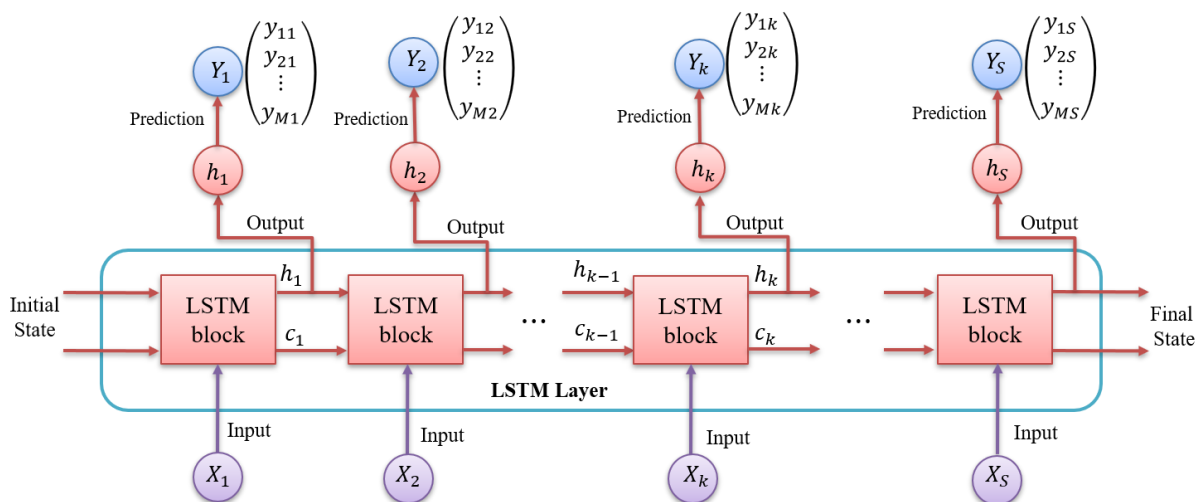


Figure 5.10 : Architecture du réseau LSTM pour la régression.

En appliquant ce réseau LSTM, les données d'entrée (X) sont les positions estimées par le FKE-robuste en section 5.4.1, et les données sorties (Y) sont les erreurs sur les positions estimées.

Pour entraîner et tester notre modèle, on doit diviser les données en trois ensembles distincts : entraîné, validation et prédiction. Pour la partie « entraîné », la Figure 5.11 présente l'architecture d'entraîné de la méthode FKER-LSTM. On détermine la position estimée par FKE-robuste en utilisant la combinaison de données GPS, Galileo et GLONASS. Le ENU_{rEKF} correspond aux positions Est, Nord et Up (Vertical) par FKE-robuste. Avant que les positions estimées n'entrent dans le modèle LSTM, on doit les normaliser pour qu'elles aient une variance moyenne et unitaire nulle. Par conséquent, les positions estimées normalisées sont maintenant les entrées de la partie « entraîné » du modèle LSTM. Les résultats cibles sont les erreurs des positions estimées par le FKE-robuste (δ_{ENU}). Les erreurs dans les positions estimées sont la différence entre les positions réelles et les positions estimées (FKE-robuste). De même, on normalise les résultats cibles pour obtenir une moyenne et une variance unitaire nulles, et les résultats cibles normalisés sont maintenant les résultats cibles de la partie « entraîné » du modèle LSTM. Le modèle est entraîné sur l'ensemble des données « entraîné » en utilisant l'apprentissage supervisé avec des méthodes d'optimisation telles que : descente de gradient stochastique (stochastic gradient descent), Adam, RMSprop. Chaque fois, le modèle LSTM est exécuté avec les entrées de la partie « entraîné » et renvoie les sorties de la partie « entraîné », qui sont ensuite comparées avec les sorties cibles. En fonction de résultats de la comparaison, les paramètres du modèle (poids et biais) sont ajustés. Le processus de la partie « entraîné » fonctionne jusqu'à ce que les erreurs de la partie « entraîné » atteignent le seuil d'erreur et le modèle LSTM inféré est appelé « bon modèle LSTM » (fitting LSTM model).

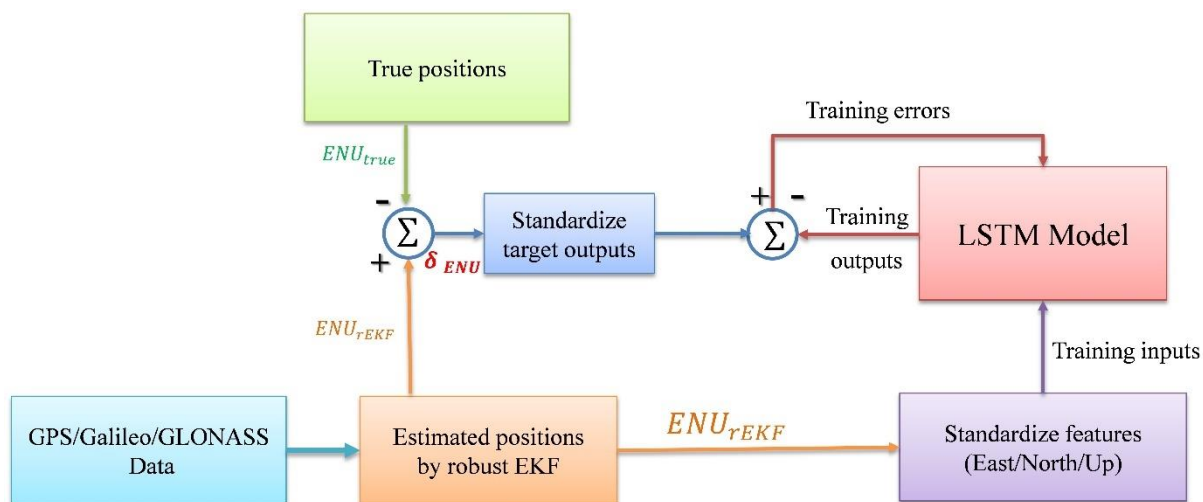


Figure 5.11 : Architecture d'entraîné de la méthode FKER-LSTM

La partie suivante concerne la partie « validation » dont l'architecture est illustrée par la Figure 5.12. Tout d'abord, on normalise les positions estimées de l'Est, du Nord et du Up (Vertical). Ensuite, on utilise le « bon modèle LSTM » de la partie « entraîné » pour prédire les erreurs standardisées de l'ENU. Et puis, on les dénormalise pour obtenir les erreurs prévues de l'ENU. Après, on compare les erreurs prédites avec les véritables erreurs de l'ENU. La comparaison nous permet de voir la qualité du modèle LSTM qui se généralise dans la partie « entraîné ». Si le modèle est trop ou peu adapté, il est nécessaire de revenir à la partie « entraîné » pour régler les hyper-paramètres du modèle (le nombre de couches, le nombre d'unités cachées dans chaque

couche, etc). On peut revenir plusieurs fois à la partie « entraîné » jusqu’à ce qu’on obtient le « meilleur modèle LSTM » pour assurer la confiance sur les performances du modèle proposé.

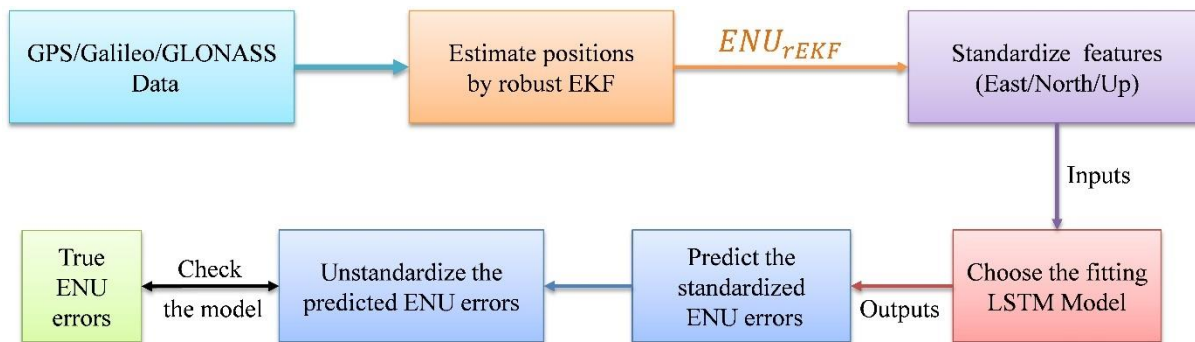


Figure 5.12 : Architecture de validation de la méthode FKER-LSTM

Après avoir obtenu le meilleur modèle LSTM, ce modèle est utilisé dans la partie « prédiction ». La Figure 5.13 présente l’architecture de prédiction de FKER-LSTM. Tout d’abord, on normalise l’estimation de l’Est, du Nord, et vers le haut (Up) similaire à la partie validation. Ensuite, on utilise le « meilleur modèle LSTM » précédemment validé dans la partie « validation », pour prévoir les erreurs de standardisation de l’ENU. Et puis, on déstandardise pour obtenir les erreurs prévues de l’ENU. Enfin, on obtient les positions estimées par FKER-LSTM, qui sont calculées comme la différence entre les positions estimées par FKE-robuste et les erreurs données par l’équation (5.126)

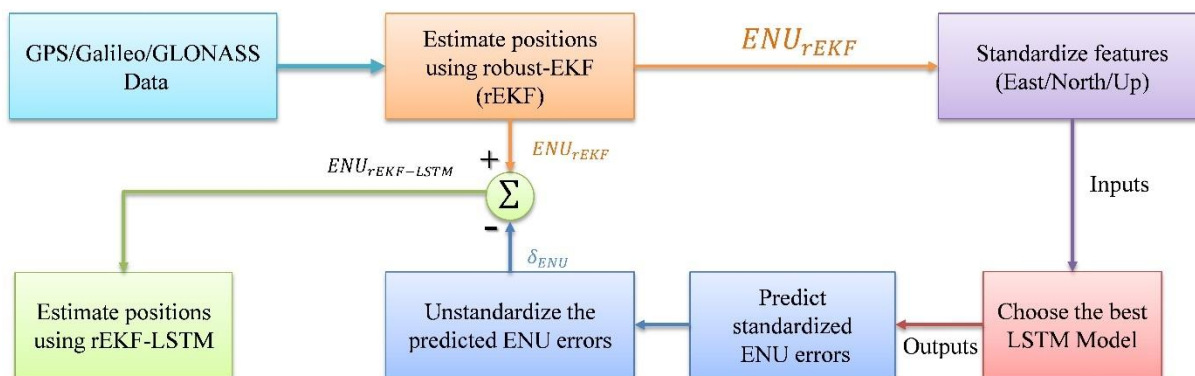


Figure 5.13 : Architecture de prédiction de la méthode FKER-LSTM

Les positions estimées par FKER-LSTM sont données par :

$$ENU_{rEKF-LSTM} = ENU_{rEKF} - \delta_{ENU} \quad (5.126)$$

5.4.2.2. Résultats de simulation

Dans cette section, on vise à comparer les performances des deux méthodes d’estimation de la position : la méthode FKE-robuste et la méthode hybride FKER-LSTM.

L’architecture LSTM est configurée de la manière suivante : une couche cachée avec $H = 20$ unités, $N = 3$ caractéristiques d’entrée (positions), $S = 1440$ nombres de pas de temps et $M = 3$ caractéristiques de sortie (position d’erreur). Les données utilisées proviennent toujours de la station ABMF enregistrée au 1^{er} janvier 2019. La Figure 5.14 représente les erreurs de position par rapport au temps pour les deux méthodes : FKE-robuste et FKER-LSTM. La période des données est de 12h à 24h, puisque les données de 0h à 12h sont utilisées pour entraîner le modèle.

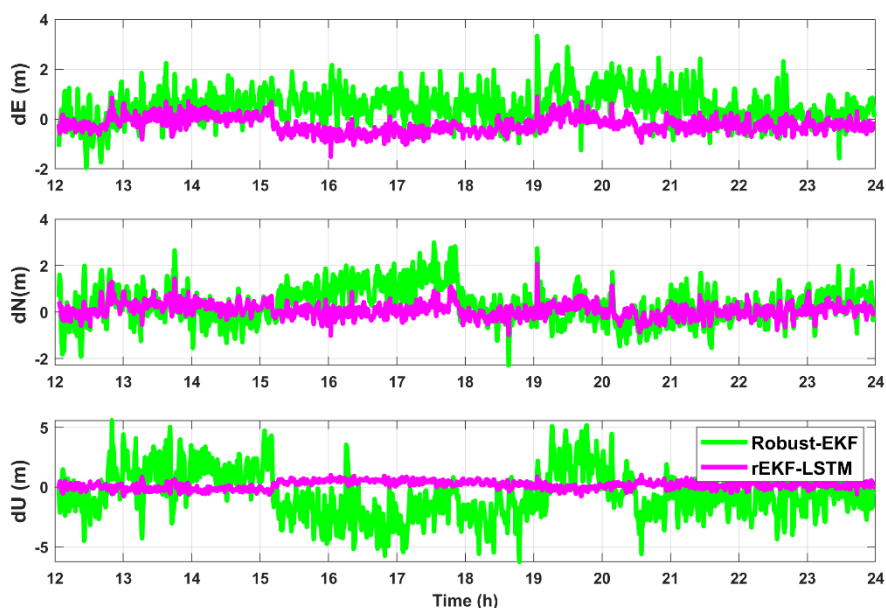


Figure 5.14 : Comparaison des erreurs de position obtenues par FKE-robuste (Robust-EKF) et FKER-LSTM (rEKF-LSTM)

Dans la Figure 5.14, la courbe en vert représente les erreurs de position obtenues avec la méthode FKE-robuste, tandis que la courbe en violet représente les erreurs produites par la méthode FKER-LSTM. On peut noter que la précision de position donnée par la méthode FKER-LSTM s'améliore largement et reste stable. Pour une analyse plus fine, le Tableau 5.9 résume les erreurs RMS de position donnée par les deux méthodes. Comme le montre ce tableau, la précision de la position s'améliore d'environ 74.0% en utilisant la méthode FKER-LSTM par rapport à l'utilisation de la méthode FKE-robuste.

Tableau 5.9 : Erreur RMS de position pour les deux méthodes à la fin 12h

Erreur RMS (m)	FKE-robuste	FKER-LSTM
RMS_E	0.8	0.4
RMS_N	0.8	0.3
RMS_U	2.0	0.4
3D-RMS	2.4	0.6

Pour avoir une vue plus générale, les cinq scénarios de la section 5.4.1 et les deux scénarios de cette section sont combinés. Par conséquent, six scénarios sont envisagés puisque le scénario relatif à FKE-robuste est commun. Les résultats sont résumés dans le Tableau 5.10.

Tableau 5.10 : Erreur RMS de position pour les six scénarios à la fin de 12h.

Erreur RMS (m)	GPS	GAL	GLO	GPS/GAL/GLO	FKE-robuste	FKER-LSTM
RMS-E	1.4	1.2	30.7	7.1	0.8	0.4
RMS-N	1.6	1.2	6.2	2.3	0.8	0.3
RMS-U	3.3	3.1	21.1	8.2	2.0	0.4
3D-RMS	4.0	3.5	37.8	11.1	2.3	0.6

Comme le montre le Tableau 5.10, la précision de la position est améliorée d'environ 95% en utilisant le méthode hybride FKER-LSTM par rapport à la combinaison de données non-robustes. Ce résultat prouve que l'application de la méthode LSTM peut améliorer considérablement la précision de la position. Afin d'évaluer plus en profondeur la méthode FKER-LSTM, d'autres données de stations de base sont testées et les résultats sont résumés dans le Tableau 5.11.

Tableau 5.11 : Erreurs RMS de position aux trois stations AJAC, GRAC et LMMF

Station de base	Erreur RMS (m)	GPS	GAL	GLO	GPS/GAL/GLO	FKE-robuste	FKER-LSTM
AJAC	RMS-E	1.1	1.4	17.8	3.4	0.7	0.7
	RMS-N	1.6	1.3	9.3	4.6	0.9	0.5
	RMS-U	2.5	2.4	3.4	4.8	1.3	0.5
	3D-RMS	3.2	3.0	20.3	7.4	1.8	1.0
GRAC	RMS-E	1.1	1.4	17.4	3.4	0.7	0.8
	RMS-N	1.6	1.3	11.9	5.9	0.9	0.5
	RMS-U	2.9	3.0	12.7	4.8	2.3	1.7
	3D-RMS	3.5	3.6	24.6	8.3	2.6	1.9
LMMF	RMS-E	1.6	1.1	33.1	6.6	0.7	0.5
	RMS-N	1.4	1.0	11.8	1.8	0.6	0.3
	RMS-U	3.4	2.2	4.3	8.2	1.8	0.5
	3D-RMS	4.0	2.6	35.4	10.7	2.0	0.8

Le Tableau 5.11 donne les erreurs RMS de positions pour les autres stations de base : AJAC, GRAC et LMMF de la fin 12h. La précision de la position est améliorée d'environ 87%, 77% et 93% en utilisant le FKER-LSTM par rapport à la combinaison de données non-robustes des trois stations de base AJAC, GRAC et LMMF en France au 1^{er} janvier 2019. Cela corrobore la conclusion selon laquelle la méthode FKER-LSTM peut améliorer considérablement la précision de la position.

5.5. Positionnement relatif

Le principe du positionnement relatif a été présenté dans la section 4.5. Le positionnement relatif est ici appliqué pour corriger les erreurs GNSS des mesures de pseudo-distance (ionosphère, troposphère, horloge du satellite et horloge du récepteur) lors du suivi d'un satellite à partir de deux récepteurs (un récepteur connu et un récepteur inconnu). Pour appliquer ce positionnement, on utilise les données GLONASS et GPS du navire de l'ENSTA Bretagne du 30 novembre 2015 avec la station de référence BRST à Brest. Les données RINEX et les éphémérides diffusées de la station BRST sont utilisées pour le calcul. Elles ont été téléchargées à partir du réseau RGP pour la journée du 30 novembre 2015. Dans ce cas, on utilise les données d'un navire statique.

5.5.1. Résultats de simulation

On applique l'algorithme de filtrage de Kalman étendu pour déterminer la position et la vitesse de l'utilisateur. L'état de l'utilisateur est défini par :

$$X = [x_r \ v_x \ y_r \ v_y \ z_r \ v_z]^T \quad (5.127)$$

Où $[x_r, y_r, z_r]$ est la position de l'utilisateur ; $[v_x, v_y, v_z]$ est la vitesse de l'utilisateur.

L'état X est composé de 6 paramètres : 3 positions et 3 vitesses, étant donné que les biais de l'horloge du récepteur et des satellites sont éliminés par la double différence.

L'algorithme de filtrage se déroule en deux étapes : prédiction et mise à jour.

Etape 1 : Prédiction

On suppose qu'il n'y a pas de couplage entre x_r, y_r, z_r . L'état de prédiction X_k^- à l'instant t_k est défini comme :

$$X_k^- = f(X_{k-1}) = \begin{bmatrix} x_{r,k-1} + \Delta T * v_x \\ v_x \\ y_{r,k-1} + \Delta T * v_y \\ v_y \\ z_{r,k-1} + \Delta T * v_z \\ v_z \end{bmatrix} \quad (5.128)$$

En linéarisant la fonction $f(X_{k-1})$, la matrice de transition d'état A_k est obtenue comme une matrice diagonale en bloc 3x3x de la forme :

$$A_k = \frac{\partial f}{\partial X} \Big|_{X=X_{k-1}} = \begin{bmatrix} A_x & 0 & 0 \\ 0 & A_y & 0 \\ 0 & 0 & A_z \end{bmatrix} \quad (5.129)$$

Où $A_x = A_y = A_z = \begin{bmatrix} 1 & \Delta T \\ 0 & 1 \end{bmatrix}$ avec ΔT l'intervalle de temps entre deux époques.

Après avoir déterminé la matrice de transition d'état A_k , on calcule la prédiction de covariance d'erreur par :

$$P_k^- = A_k P_{k-1} A_k^T + Q_k \quad (5.130)$$

Où

$$Q_k = \text{digonal}(Q_x, Q_y, Q_z) \quad (5.131)$$

avec $Q_x = \begin{bmatrix} S_x \Delta T + S_{vx} (\Delta T)^3 / 3 & S_{vx} (\Delta T)^2 / 2 \\ S_{vx} (\Delta T)^2 / 2 & S_{vx} \Delta T \end{bmatrix}$, Q_y et Q_z sont déterminées de la même manière que pour Q_x ; S_x, S_{vx} sont les densités spectrales de puissance du bruit de position et de vitesse dans les coordonnées X.

Etape 2 : Mise à jour

L'équation d'observation est définie par : $Z_k = h(X_{k-1}) + e_k$, avec

$$Z_k = [Z_G \ Z_R]^T \quad (5.132)$$

Z_G représente les pseudo-distances corrigées et les taux de pseudo-distances corrigées en double différence de GPS, respectivement au temps t_k . Ici, on considère le premier satellite qui est le satellite référence :

$$Z_G = \begin{bmatrix} \rho_G^2 - \rho_G^1 \\ \vdots \\ \rho_G^{N_{gps}} - \rho_G^1 \\ \dot{\rho}_G^2 - \dot{\rho}_G^1 \\ \vdots \\ \dot{\rho}_G^{N_{gps}} - \dot{\rho}_G^1 \end{bmatrix} \quad (5.133)$$

Où ρ_G^i est la pseudo-distance du $i^{\text{ème}}$ satellite et $\dot{\rho}_G^i$ est le taux de variation de pseudo-distance du $i^{\text{ème}}$ satellite avec $i = 1, \dots, N_{gps}$; N_{gps} est nombre de satellites GPS suivis.

Z_R correspond aux pseudo-distances corrigées et les taux de pseudo-distances corrigées en double différence de GLONASS, respectivement au temps t_k . Ici, on considère le premier satellite qui est le satellite référence :

$$Z_R = \begin{bmatrix} \rho_R^2 - \rho_R^1 \\ \vdots \\ \rho_R^{N_{glo}} - \rho_R^1 \\ \dot{\rho}_R^2 - \dot{\rho}_R^1 \\ \vdots \\ \dot{\rho}_R^{N_{glo}} - \dot{\rho}_R^1 \end{bmatrix} \quad (5.134)$$

Où ρ_R^i est la pseudo-distance du $i^{\text{ème}}$ satellite et $\dot{\rho}_R^i$ est le taux de la pseudo-distance du $i^{\text{ème}}$ satellite avec $i = 1, \dots, N_{glo}$; N_{glo} est le nombre de satellites GLONASS suivis.

La matrice H_k est déterminée par dérivée de la fonction $h(X_k^-)$: $H_k = \left. \frac{\partial h}{\partial X} \right|_{X=X_k^-}$

$$H_k = [H_{k,G} \ H_{k,R}]^T \quad (5.135)$$

Où $H_{k,G}$ et $H_{k,R}$ sont les Jacobiens du modèle de mesure avec les données GPS et GLONASS, respectivement qui sont calculés comme :

$$H_{k,G} = \begin{bmatrix} a_x^2 - a_x^1 & 0 & a_y^2 - a_y^1 & 0 & a_z^2 - a_z^1 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_x^{N_{gps}} - a_x^1 & 0 & a_y^{N_{gps}} - a_y^1 & 0 & a_z^{N_{gps}} - a_z^1 & 0 \\ 0 & a_x^2 - a_x^1 & 0 & a_y^2 - a_y^1 & 0 & a_z^2 - a_z^1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & a_x^{N_{gps}} - a_x^1 & 0 & a_y^{N_{gps}} - a_y^1 & 0 & a_z^{N_{gps}} - a_z^1 \end{bmatrix} \quad (5.136)$$

Où

$$\begin{aligned} a_x^i &= \frac{\partial \rho_G^i}{\partial x} = \frac{-(x_i - \hat{x}_r^-)}{\sqrt{(x_i - \hat{x}_r^-)^2 + (y_i - \hat{y}_r^-)^2 + (z_i - \hat{z}_r^-)^2}} \\ a_y^i &= \frac{\partial \rho_G^i}{\partial y} = \frac{-(y_i - \hat{y}_r^-)}{\sqrt{(x_i - \hat{x}_r^-)^2 + (y_i - \hat{y}_r^-)^2 + (z_i - \hat{z}_r^-)^2}} \\ a_z^i &= \frac{\partial \rho_G^i}{\partial z} = \frac{-(z_i - \hat{z}_r^-)}{\sqrt{(x_i - \hat{x}_r^-)^2 + (y_i - \hat{y}_r^-)^2 + (z_i - \hat{z}_r^-)^2}} \end{aligned} \quad (5.137)$$

$$H_{k,R} = \begin{bmatrix} b_x^2 - b_x^1 & 0 & b_y^2 - b_y^1 & 0 & b_z^2 - b_z^1 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ b_x^{N_{glo}} - b_x^1 & 0 & b_y^{N_{glo}} - b_y^1 & 0 & b_z^{N_{glo}} - b_z^1 & 0 \\ 0 & b_x^2 - b_x^1 & 0 & b_y^2 - b_y^1 & 0 & b_z^2 - b_z^1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & b_x^{N_{glo}} - b_x^1 & 0 & b_y^{N_{glo}} - b_y^1 & 0 & b_z^{N_{glo}} - b_z^1 \end{bmatrix} \quad (5.138)$$

Où

$$\begin{aligned} b_x^j &= \frac{\partial \rho_R^j}{\partial x} = \frac{-(x_j - \hat{x}_r^-)}{\sqrt{(x_j - \hat{x}_r^-)^2 + (y_j - \hat{y}_r^-)^2 + (z_j - \hat{z}_r^-)^2}} \\ b_y^j &= \frac{\partial \rho_R^j}{\partial y} = \frac{-(y_j - \hat{y}_r^-)}{\sqrt{(x_j - \hat{x}_r^-)^2 + (y_j - \hat{y}_r^-)^2 + (z_j - \hat{z}_r^-)^2}} \\ b_z^j &= \frac{\partial \rho_R^j}{\partial z} = \frac{-(z_j - \hat{z}_r^-)}{\sqrt{(x_j - \hat{x}_r^-)^2 + (y_j - \hat{y}_r^-)^2 + (z_j - \hat{z}_r^-)^2}} \end{aligned} \quad (5.139)$$

Après avoir déterminé la matrice H_k , le gain de Kalman est calculé par :

$$K = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \quad (5.140)$$

Pour simplifier le calcul, on met la matrice R_k sous la forme :

$$R_k = I * R_{erreur} \quad (5.141)$$

Où I est la matrice identité avec la dimension $2*(N_{gps}+N_{glo}-2)$, et $R_{erreur} = 0.1$ (m) (après plusieurs cas testés de R_{erreur}).

Pour évaluer les performances de l'algorithme, deux scénarios sont considérés et les simulations ont été effectuées sous l'environnement Matlab :

Scénario # 1: solution de navigation basée sur les données GLONASS,

Scénario # 2: solution de navigation basée sur les données GLONASS et GPS.

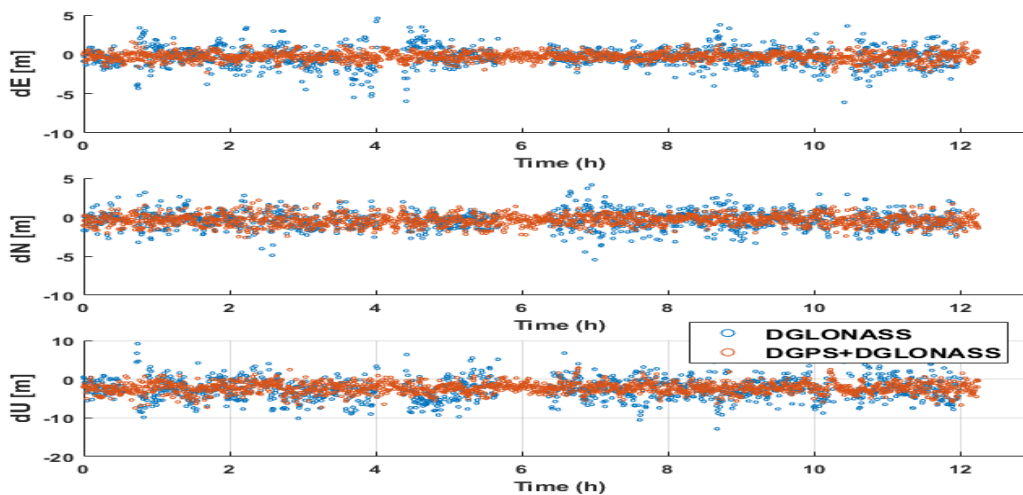


Figure 5.15 : Comparaison des erreurs dans les estimations de position

La Figure 5.15 montre les erreurs de position du récepteur dans les deux scénarios en utilisant une double différence. La courbe en bleu représente le scénario #1 avec uniquement des données GLONASS ; les résultats sont instables. Surtout, autour de 4h et 6h, les données GLONASS ne permettent pas de déterminer la position. La combinaison des données GLONASS et GPS résout ces problèmes - comme illustré sur la courbe en rouge du scénario #2. Une synthèse des résultats est donnée dans le Tableau 5.12 pour les deux scénarios considérés.

Tableau 5.12 : Erreur RMS en position

Erreur RMS (m)	DGLONASS	DGLONASS+ DGPS
RMS_E	1.3	0.6
RMS_N	1.2	0.8
RMS_U	3.8	2.7
3D-RMS	4.2	2.9

Tableau 5.13 : Erreur Max, Min et l'écart-type

Erreur (m)	Scénario #	
	1	2
Max_E	6.1	2.3
Min_E	5.2e-04	12e-04
Ecart type E	0.9	0.4
Max_N	5.5	2.6
Min_N	12e-04	2.8e-04
Ecart type N	0.7	0.5
Max_U	12.8	7.5
Min_U	59e-04	66e-04
Ecart type U	2.1	1.3

Selon les valeurs « Erreur RMS » dans le Tableau 5.12, les données différentielles GLONASS et GPS améliorent la précision de la position d'environ 31% par rapport à l'utilisation des données GLONASS uniquement.

5.6. Conclusion

Dans ce chapitre, nous avons présenté les algorithmes : des « moindres carrés itérativement pondérés » et le « filtre de Kalman étendu robuste » pour le mode de positionnement absolu avec la combinaison des données GPS, Galileo et GLONASS. Tandis que pour le mode de positionnement, nous avons présenté l'algorithme « filtre de Kalman étendu » avec la combinaison des données GPS et GLONASS prise sur le bateau de l'ENSTA Bretagne.

Ce chapitre met en avant le mode de positionnement absolu, car ses avantages sont le faible coût et le traitement simple des données. La fusion de données peut contenir des valeurs aberrantes qui ont un impact négatif sur les résultats finaux. D'autres algorithmes ont alors été présentés : les « moindres carrés itérativement repondérés » et le « filtre de Kalman étendu robuste ». En particulier, nous avons montré que l'algorithme « filtre de Kalman étendu robuste » est plus efficace. Il est construit en combinant le filtrage de Kalman étendu, le MM-Estimation, l'IRWLS et la méthode RAIM pour améliorer la précision de position. La méthode MM-Estimation est utilisée pour éliminer les valeurs aberrantes ; la méthode IRWLS est utilisée pour optimiser les résultats estimés et la méthode RAIM est utilisée pour détecter les erreurs de positionnement dépassant la limite d'alerte. La fusion robuste des données GPS, Galileo et GLONASS à l'aide de la méthode FKE-robuste augmente la précision de la position d'environ 84.0% par rapport à une estimation non-robuste en utilisant les données de la station de base ABMF. En outre, la précision de la position est considérablement améliorée à l'aide de la méthode FKER-LSTM d'environ 95% par rapport à la combinaison de données non-robuste. Pour illustrer la puissance de la méthode hybride FKER-LSTM, l'approche est utilisée pour les données de trois autres stations de base : AJAC, GRAC et LMMF. La précision de la position est largement améliorée d'environ 87%, 77% et 93% par rapport aux combinaisons de données non-robustes, respectivement.

Dans ce travail, l'accent est mis sur l'amélioration de la précision du positionnement pour un récepteur se déplaçant lentement ; seules les données des stations de base sont utilisées. Dans le chapitre suivant, on va présenter la combinaison des données GNSS et des données INS pour des utilisateurs en mouvement.