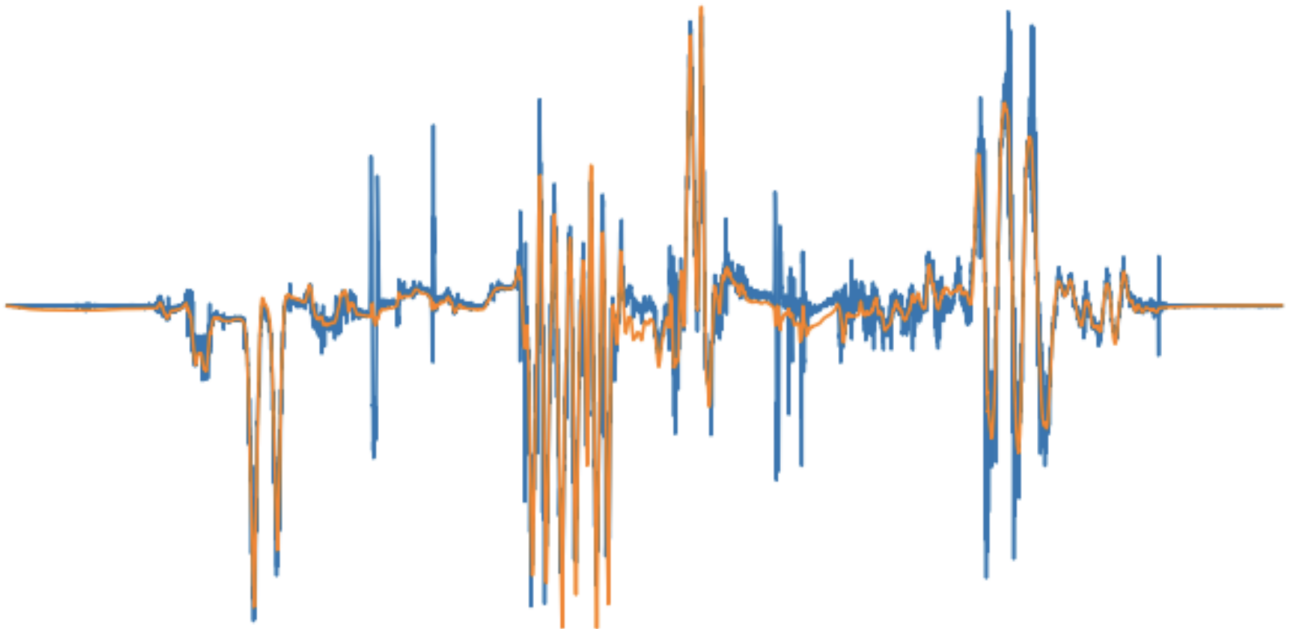


TEST_ALGO_NAV_V1

Zayed Herma



1 CONTEXTE ET OBJECTIF

L'objectif est de mettre en œuvre un algorithme d'observation non linéaire de type "filtre complémentaire", qui estime l'attitude à partir de données capteurs accéléromètres, gyromètres et magnétomètres, pour pouvoir mieux estimer l'orientation (angles de roulis, tangage, lacet) à partir de données réelles obtenues par un système Sysnav.

Le filtre implémenté est celui décrit dans l'article de Hamel et Mahony (Attitude estimation on SO(3) based on direct inertial measurements, ICRA 2170-2175, IEEE (2006))

2 Description de la méthode suivie pour implémenter le filtre

2.1 Estimation de l'orientation à partir uniquement des mesures

Les mesures utilisées pour déterminer la matrice d'orientation R sont celles fournies par l'accéléromètre et le magnétomètre.

La matrice d'orientation théorique R vérifie à chaque instant les équations suivantes :

$$a = -R^T(g - \frac{dv}{dt}) \quad (1)$$

$$m = R^T m_0 \quad (2)$$

- a : l'accélération exprimée dans le repère local (repère des mesures).
- g : la pesanteur exprimée dans le repère fixe (North-East-Down).
- $\frac{dv}{dt}$: l'accélération du repère local.
- m : le vecteur champ magnétique terrestre exprimé dans le repère local.
- m_0 : le vecteur champ magnétique terrestre exprimé dans le repère fixe (North-East-Down).

NB : On suppose par la suite que $\|\frac{dv}{dt}\| \ll \|g\|$ et que le repère local est orthonormé (pour la justification voir la fin du Jupyter Notebook).

2.1.1 Estimation de g et m_0

Dans la littérature, on peut trouver les valeurs de g et m_0 car elles sont peu variables dans le temps et l'espace au voisinage de la terre. Mais, dans notre cas, nous pouvons donner une meilleure estimation à partir des données fournies par les capteurs.

Dans le repère fixe (North-East-Down), on a :

$$g = \begin{pmatrix} 0 \\ 0 \\ g_0 \end{pmatrix} \quad (3)$$

$$m_0 = \begin{pmatrix} B_h \\ 0 \\ B_v \end{pmatrix} \quad (4)$$

- g_0 : la norme de la pesanteur.
- $B_h = B \cos(I)$: la composante horizontale du champ magnétique
- $B_v = B \sin(I)$: la composante verticale du champ magnétique.
- B : la norme du vecteur champ magnétique terrestre.
- I : l'inclinaison magnétique.

avec :

$$g_0 = \sum_i^N \frac{\|a_i\|}{N} \quad (5)$$

$$B = \sum_i^N \frac{\|m_i\|}{N} \quad (6)$$

Plus le nombre N de mesures fournies par l'accéléromètre et le magnétomètre est important, meilleure est l'estimation.

Enfin pour estimer l'angle d'inclinaison I , le World Magnetic Model 2010 (valable entre 2014 et 2015) a été utilisé pour fournir une valeur de I pour l'année 2014 à Vernon, car les mesures sont effectuées en 2014.

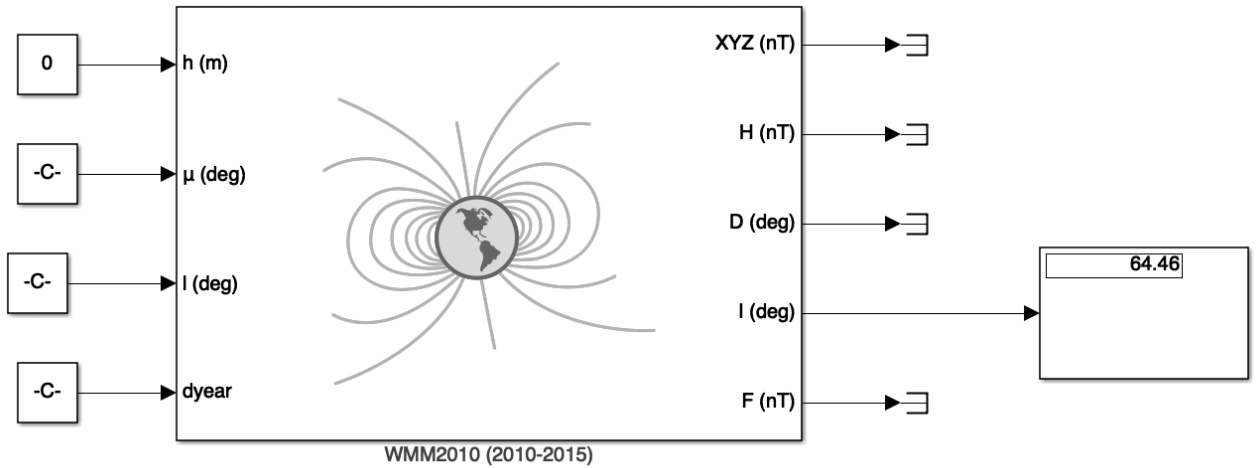


FIGURE 1 – Modèle Simulink pour estimer I

2.1.2 Estimation de R

Pour estimer la valeur empirique de R , il a été proposé dans l'article de résoudre le problème de minimisation suivant :

$$\arg \min_{R \in SO(3)} (\lambda_1 \|g + Ra\| + \lambda_2 \|m_0 - Rm\|) \quad (7)$$

Avec λ_1 et λ_2 sont choisis en fonction de la confiance dans les mesures de l'accéléromètre et du magnétomètre.

Pour résoudre ce problème de minimisation de manière simple, il a été choisi de se faire entièrement confiance aux mesures fournies par l'accéléromètre par rapport aux mesures fournies par le magnétomètre ($\lambda_1 = +\infty$), car le moindre déplacement de courant pour faire fonctionner des composants électroniques dans un système qui contient un magnétomètre perturbe les mesures fournies par ce dernier.

Si on note $R = \begin{pmatrix} r_1 & r_2 & r_2 \\ r_3 & r_4 & r_5 \\ r_6 & r_7 & r_8 \end{pmatrix}$, la résolution du problème précédent est finalement réduite à la réalisation des trois étapes suivantes :

1. Résoudre $a = -R^T g$ pour obtenir $(r_6 \ r_7 \ r_8)$:

$$-g_0 \begin{pmatrix} r_6 & r_7 & r_8 \end{pmatrix} = \begin{pmatrix} a_1 & a_2 & a_3 \end{pmatrix} \quad (8)$$

2. Résoudre le problème de minimisation simplifié suivant, pour obtenir $(r_1 \ r_2 \ r_3)$:

$$\arg \min_{\begin{pmatrix} r_1 \\ r_2 \\ r_3 \end{pmatrix} \in \mathbb{R}^3, r_6 r_1 + r_7 r_2 + r_8 r_3 = 0} \left\| B_h \begin{pmatrix} r_1 \\ r_2 \\ r_3 \end{pmatrix} + B_v \begin{pmatrix} r_6 \\ r_7 \\ r_8 \end{pmatrix} - \begin{pmatrix} m_1 \\ m_2 \\ m_3 \end{pmatrix} \right\|^2 \quad (9)$$

C'est un problème de minimisation simple, car la fonction objectif est **α -convexe continue** sur l'ensemble $K = \left\{ (r_1 \ r_2 \ r_3) \in \mathbb{R}^3, r_6 r_1 + r_7 r_2 + r_8 r_3 = 0 \right\}$ qui est **convexe fermé**. Ainsi, la théorie de l'existence d'un minimum, **cas fortement convexe**, nous assure l'existence et l'unicité du minimum. Parmi les algorithmes qui peuvent être utilisés pour résoudre efficacement ce problème de minimisation figure **l'algorithme du gradient à pas fixe avec projection** qui, selon la théorie, converge dans notre cas **fortement convexe**.

Dans l'implémentation, on a utilisé **scipy.optimize.LinearConstraint**, qui fait appel à la programmation séquentielle des moindres carrés (SLSQP).

Enfin, on normalise la solution optimale obtenue pour respecter toujours la contrainte initiale $R \in SO(3)$.

$$\begin{pmatrix} r_1 & r_2 & r_3 \end{pmatrix} = \frac{\begin{pmatrix} r_1^* & r_2^* & r_3^* \end{pmatrix}}{\left\| \begin{pmatrix} r_1^* & r_2^* & r_3^* \end{pmatrix} \right\|} \quad (10)$$

Cela peut avoir pour conséquence d'affaiblir l'optimalité de la solution de (9). Par conséquent, une autre méthode plus efficace consiste à résoudre le problème d'optimisation précédent, mais pénalisé par une contrainte supplémentaire :

$$\arg \min_{\begin{pmatrix} r_1 \\ r_2 \\ r_3 \end{pmatrix} \in \mathbb{R}^3, r_6 r_1 + r_7 r_2 + r_8 r_3 = 0} \left(\alpha_1 \left\| B_h \begin{pmatrix} r_1 \\ r_2 \\ r_3 \end{pmatrix} + B_v \begin{pmatrix} r_6 \\ r_7 \\ r_8 \end{pmatrix} - \begin{pmatrix} m_1 \\ m_2 \\ m_3 \end{pmatrix} \right\|^2 + \alpha_2 \left(\left\| \begin{pmatrix} r_1 \\ r_2 \\ r_3 \end{pmatrix} \right\|^2 - 1 \right)^2 \right) \quad (11)$$

Avec le choix de α_1 et α_2 dépendant de la confiance dans les mesures fournies par le magnétomètre et de la sensibilité du filtre à la contrainte $R \in SO(3)$.

3. Pour trouver $(r_4 \ r_5 \ r_6)$ en respectant la contrainte $R \in SO(3)$, on calcule le produit vectoriel :

$$(r_4 \ r_5 \ r_6) = (r_4 \ r_5 \ r_6) \times (r_1 \ r_2 \ r_3) \quad (12)$$

A l'issue de ces trois étapes, on arrive à une estimation empirique de R , basée uniquement sur les mesures de l'accéléromètre et du magnétomètre, qui est de mieux en mieux si les mesures sont peu bruitées.

2.2 L'algorithme du filtre passif complémentaire

D'après l'article et dans notre cas particulier où le repère fixe (North-East-Down) est orthonormé, l'estimateur d'orientation \hat{R} et du biais \hat{b} obéit aux équations suivantes :

$$\dot{\hat{R}} = \hat{R}\hat{\Omega}_\times \quad (13)$$

$$\dot{\hat{b}} = -\frac{\kappa_I}{\kappa_P}\omega \quad (14)$$

Avec et selon les notations de l'article :

- $\omega = \kappa_P \text{vex}(\pi_a(\tilde{R}))$: le terme d'innovation qui est 0 si le filtre est conforme aux mesures.
- $\tilde{R} = \hat{R}^T R$.
- $\pi_a(\tilde{R}) = \frac{\tilde{R} - \tilde{R}^T}{2}$.
- $\hat{\Omega}_\times = (\Omega + b - \hat{b} + \omega)_\times$.
- $\Omega + b$: la vitesse angulaire fournie par le gyroscope qui est généralement biaisée par un terme de biais (b).

Enfin, l'erreur entre R et \hat{R} est définie par :

$$E_{mes} = 3 - \text{tr}(\tilde{R}) \quad (15)$$

2.2.1 Discrétisation de l'algorithme

Après discrétisation des équations (13) et (14), on obtient le modèle en temps discret du filtre :

- $\hat{R}^{k+1} = \hat{R}^k \hat{A}^k$
- $\hat{A}^k = \exp(dt\hat{\Omega}_\times^k) = I + \hat{\Omega}_\times^k \frac{\sin(dt|\hat{\Omega}_\times^k|)}{|\hat{\Omega}_\times^k|} + (\hat{\Omega}_\times^k)^2 \frac{1 - \cos(dt|\hat{\Omega}_\times^k|)}{(|\hat{\Omega}_\times^k|)^2}$
- $\hat{b}^{k+1} = \hat{b}^k - dt\kappa_I \text{vex}(\pi_a(\tilde{R}^k))$

Pour mettre en œuvre l'algorithme discrétisé, il faut un état initial de départ (\hat{R}^0, \hat{b}^0) qui respecte les conditions suivantes :

$$E_{mes}(0) < 6 \quad (16)$$

$$\kappa_I > \frac{|b(0) - \hat{b}(0)|}{6 - 2E_{mes}(0)} \quad (17)$$

Pour l'implémentation, on a choisi l'état initial $\hat{R}^0 = R$ et $\hat{b}^0 = 0$, de sorte que $E_{mes}(0) = 0$ pour vérifier (16) et motiver le choix du paramètre κ_I par rapport à nos connaissances de l'ordre de grandeur du biais b pour vérifier (18) :

$$\kappa_I > \frac{|b(0)|}{6} \quad (18)$$

3 Test du filtre sur des données simulées

Afin de tester les performances du filtre, des données ont été générées pour être utilisées afin de juger les performances du filtre.

3.0.1 Génération de données

Pour générer les données, on commence par générer les angles de roulis, de tangage et de lacet pour calculer la matrice R qui sera utilisée pour simuler les mesures de l'accéléromètre, du magnétomètre et du gyroscope.

Par la suite, on décrit les étapes de l'algorithme de génération de données :

1. Génération de roll, pitch et yaw à un instant k grâce à des fonctions continues f , g et h :

$$roll^k = f(k) \quad (19)$$

$$pitch^k = g(k) \quad (20)$$

$$yaw^k = h(k) \quad (21)$$

2. Génération de R_k à partir de $roll^k$, $pitch^k$ et yaw^k .
3. Génération des mesures a_k de l'accéléromètre, m_k du magnétomètre et Ω_k du gyroscope :

$$a^k = -R_k^T g \quad (22)$$

$$m^k = R_k^T m_0 \quad (23)$$

$$\Omega_{\times}^k = R_k^T \frac{R_{k+1} - R_k}{dt} \quad (24)$$

Afin de tester la résistance du filtre, on peut ajouter du bruit aux données simulées et tester sa robustesse.

3.0.2 Validation du filtre sur des données simulées sans bruit

Dans ce cas, le R filtré et mesuré coïncident avec les données réelles générées comme montré par les figures suivants :

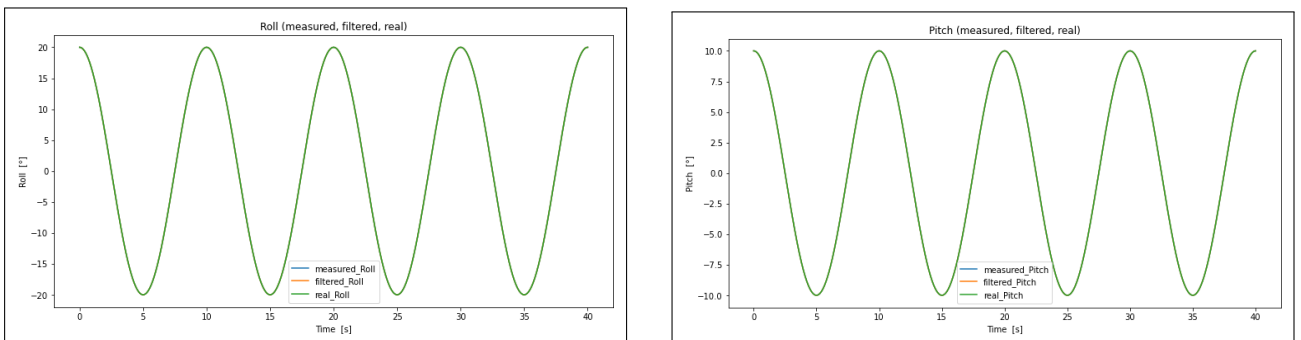


FIGURE 2 – Les angles de roulis et de tangage mesurés, filtrés et réels

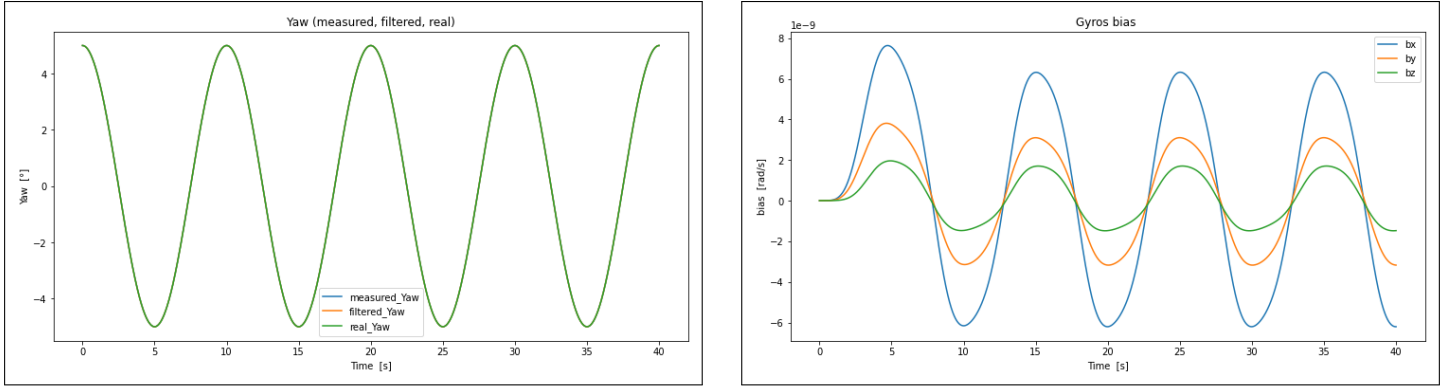


FIGURE 3 – L'angle de lacet mesuré, filtré et réel avec le biais

On constate que le biais estimé par le filtre n'est pas complètement nul : c'est dû à la procédure de discrétisation.

3.0.3 Validation du filtre sur des données simulées avec bruit

Le bruit qui a été choisi, pour les mesures simulées de l'accéléromètre et du magnétomètre, est un bruit additif du type :

$$\text{Gaussien process} \times \text{Bernoulli process} \quad (25)$$

NB : le processus de bernoulli de paramètre p est pour dire que parfois nous disposons de bonnes mesures (si $p = 0$, les mesures sont bonnes tout le temps et si $p = 1$, les mesures sont mauvaises tout le temps).

Ainsi, pour l'accéléromètre, il est commode de choisir p proche de 0 pour dire que les mesures fournies par l'accéléromètre sont peu bruitées et de choisir p proche de 1 pour dire que les mesures fournies par le magnétomètre sont très bruitées.

Pour ce qui concerne les mesures simulées du gyroscope, il a été choisi d'ajouter un biais (b_t) qui dépend du temps, en plus de l'addition d'un bruit gaussien qui est toujours présent.

On obtient pour les résultats de la simulation avec bruit et avec les paramètres ($\kappa_P = 1$, $\kappa_I = 0.3$) :

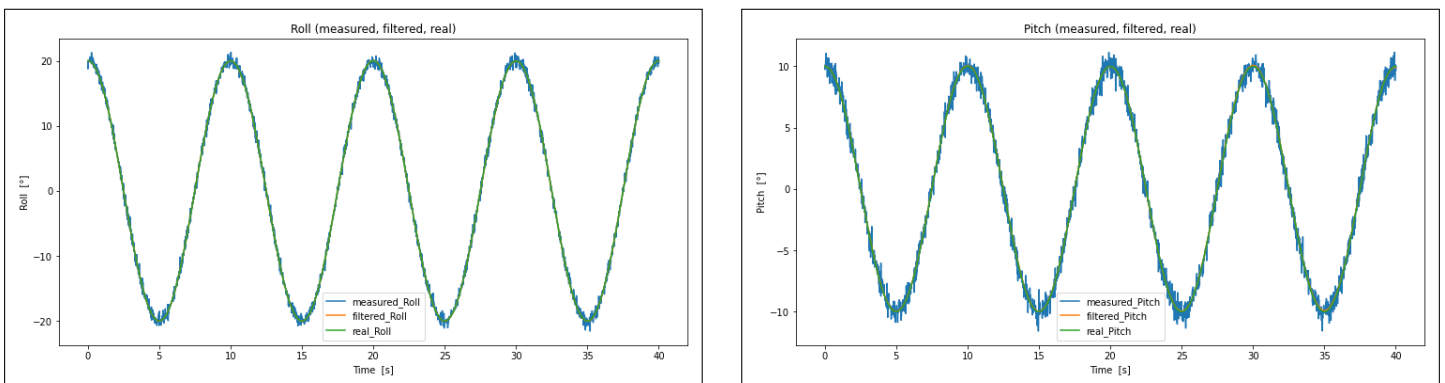


FIGURE 4 – Les angles de roulis et de tangage mesurés, filtrés et réels

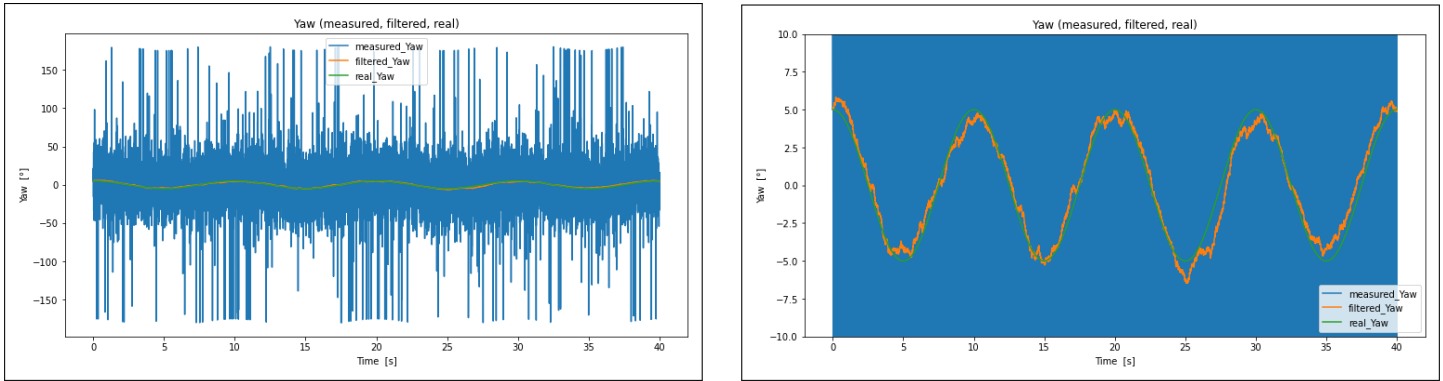


FIGURE 5 – L'angle de lacet mesuré, filtré et réel

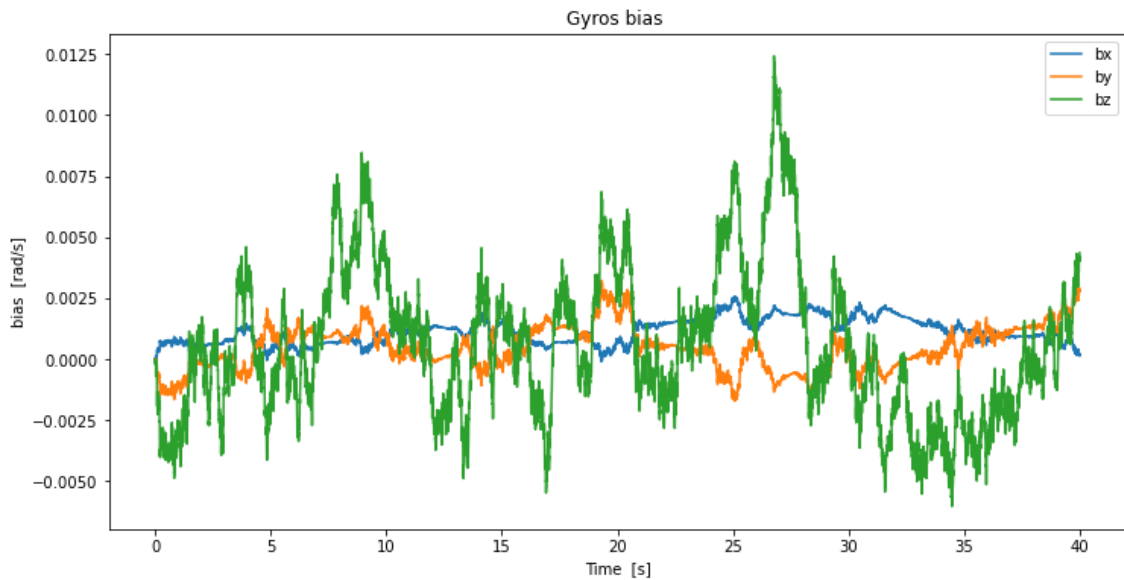


FIGURE 6 – Estimation du biais par le filtre

Nous pouvons voir que nous avons une bonne mesure des angles de roulis et de tangage et une bonne estimation par le filtre, car les mesures de l'accéléromètre sont peu bruitées. Cependant, la mesure de l'angle de lacet est très bruitée, à cause des fausses mesures du magnétomètre, contrairement à ce qui est estimé par le filtre qui est presque proche de l'angle réel.

Le filtre donne également une bonne estimation du biais, qui a le même ordre de grandeur que le bruit puisque ce dernier est plus important que le biais dans notre cas.

Après avoir testé le filtre sur différentes données simulées et vérifié qu'il se comporte correctement en fonction du niveau de bruit sur les mesures fournies par les différents capteurs, on peut conclure que ce filtre a un bon comportement général si les mesures de l'accéléromètre ne sont pas trop bruitées.

4 Les Résultats sur les données Sysnav 2014

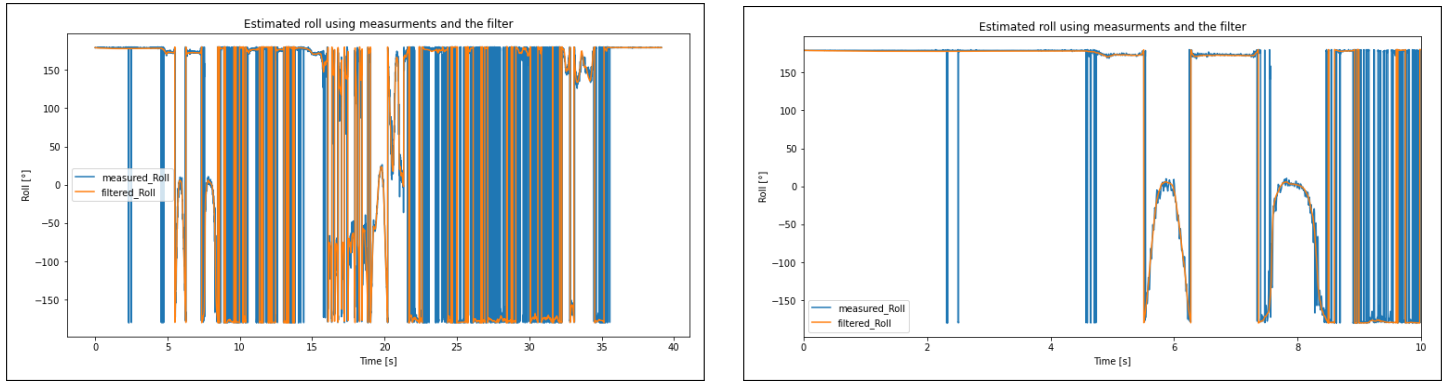


FIGURE 7 – L'angle de roulis mesuré et filtré

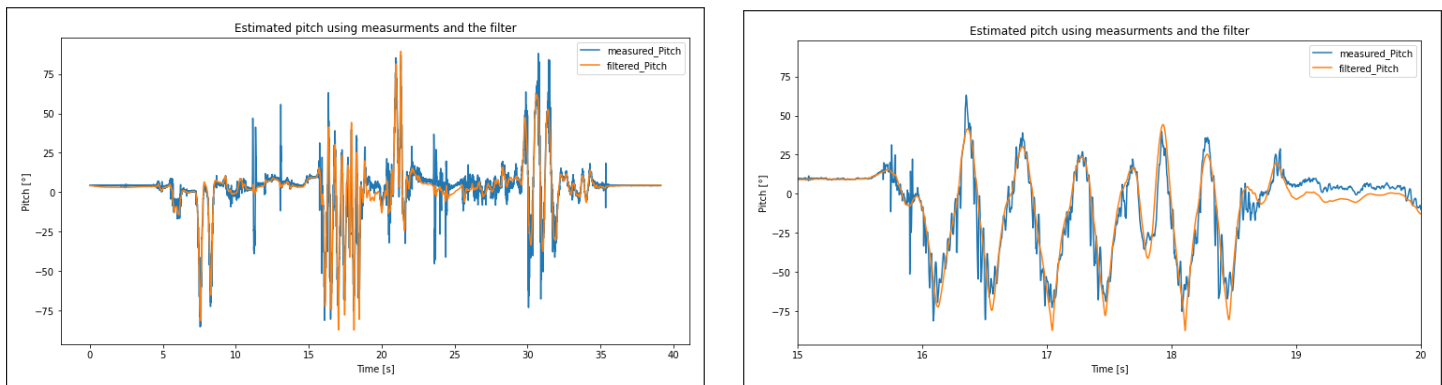


FIGURE 8 – L'angle de tangage mesuré et filtré

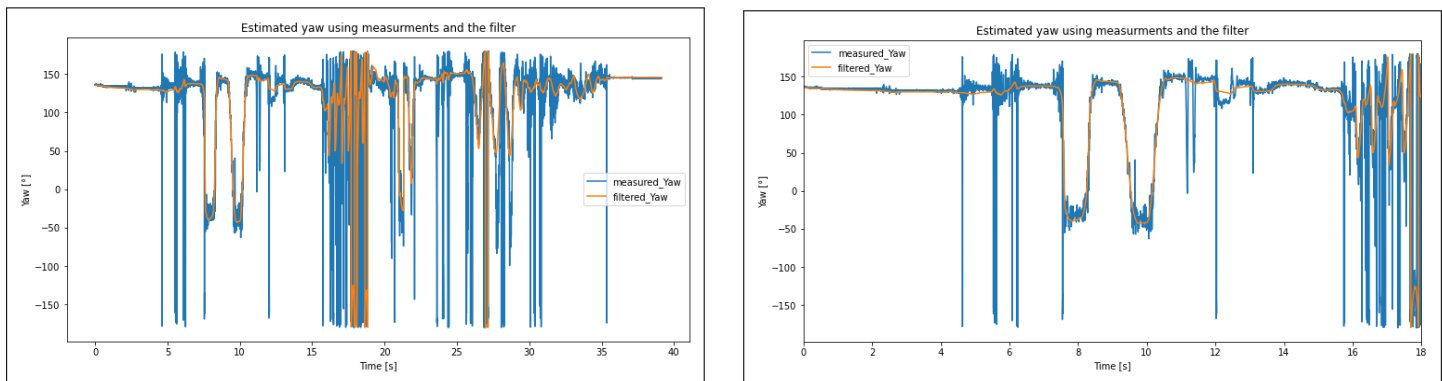


FIGURE 9 – L'angle de lacet mesuré et filtré

Le filtre se comporte bien pour estimer l'orientation, car on voit qu'il ignore les sauts brusques des mesures à faible durée. De plus, on remarque qu'il ne présente pas d'anomalies de divergences, chose à laquelle on peut parfois être confronté quand on utilise des données réels.

Ainsi, sur les données réels, le filtre suit globalement les mesures et en donne une estimation plus lisse, moins bruitée et plus exploitable. On peut donc conclure que l'implémentation est bien valide.

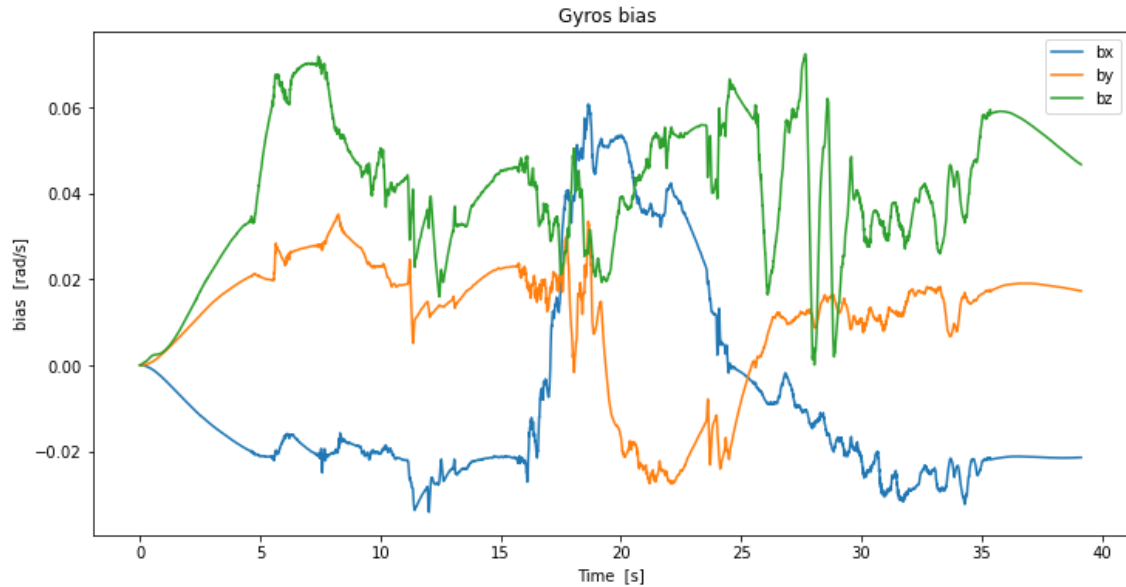


FIGURE 10 – Estimation du biais par le filtre

On remarque que le biais estimé par le filtre est du même ordre de grandeur que la vitesse angulaire mesuré par le gyroscope (autour de 0.06 rad/s). Il s'avère donc que ce biais est très élevé.

5 Discription du fichier du code

Dans le fichier de code, vous trouverez à la racine :

1. `measured_R.py` : contient une implémentation de ce qui a été décrit dans la partie 2.1 de ce document.
2. `to_filter_R.py` : contient quelques fonctions utiles pour l'implémentation de l'algorithme discrétisé décrit de manière partielle dans la partie 2.2 de ce document.
3. `estimated_roll_pitch_yaw.py` : permet de charger les données à traiter afin de sauvegarder les données mesurées et filtrées dans le dossier "Measured_Roll_Pitch_Yaw".
4. `simulated_R.py` : contient une implémentation de ce qui a été décrit dans la partie 3.0.1 de ce document.
5. `estimated_roll_pitch_yaw.py` : permet de générer des données simulées et de sauvegarder les données mesurées et filtrées dans le dossier "Simulated_Roll_Pitch_Yaw" pour juger la performance du filtre.
6. `plots.ipynb` : permet d'accéder aux dossiers pour visualiser les données qui y sont stockées.
7. le dossier "Others" : contient le fichier Simulink mentionné dans la section 2.1.1.