

# Estimation d'État par Filtre de Kalman Étendu

## pour Planeur Largué depuis Ballon Stratosphérique

Système de Navigation Inertielle (INS)

TRAN Pierre-Minh

16/01/2026

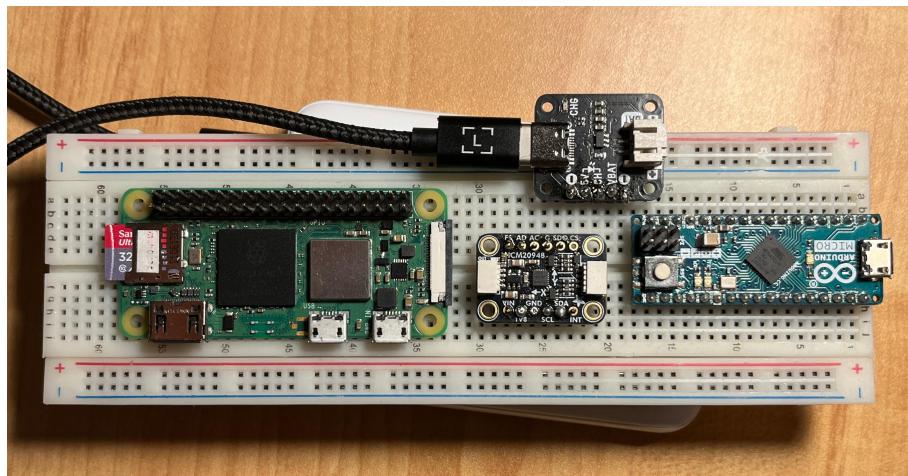


FIGURE 1 – Cartes utilisé pour la fusion des données

## 1 Introduction

Ce rapport présente la conception d'un Système de Navigation Inertielle (INS) destiné à un planeur largué depuis la haute atmosphère. L'enjeu est de maintenir une estimation précise de l'attitude (orientation), de la position et de la vitesse de l'appareil tout au long de sa descente.

Le cœur algorithmique de ce système repose sur un **Filtre de Kalman Étendu (EKF)**. Ce choix se justifie par la nécessité de fusionner des données hétérogènes provenant de capteurs aux caractéristiques variées : une centrale inertielle (IMU) à haute fréquence (100 Hz) et un GPS à plus basse fréquence (10 Hz). De plus, l'EKF permet d'estimer en temps réel les dérives (biais) des capteurs, un point crucial étant donné les variations thermiques extrêmes subies lors d'un vol stratosphérique (passant de -60°C à +20°C), qui impactent directement la précision des mesures.

## 2 Principe du Filtre de Kalman Étendu

Avant de détailler les équations, il est utile de comprendre l'EKF comme un cycle itératif en deux étapes :

1. **La Prédition (Propagation)** : En utilisant les lois de la physique (modèle dynamique) et les mesures de l'IMU, le filtre "devine" l'état suivant du planeur. Cependant, l'incertitude augmente à chaque calcul car les erreurs de mesure s'accumulent (dérive inertie).
2. **La Correction (Mise à jour)** : Dès qu'une mesure de référence est disponible (comme un signal GPS ou la détection de la gravité), le filtre compare sa prédition avec la réalité observée. Il calcule alors un compromis optimal, appelé **Gain de Kalman**, pour corriger son estimation et réduire son incertitude.

L'aspect "**Étendu**" signifie que le filtre utilise des outils mathématiques (Jacobiennes) pour linéariser les équations de vol, car les rotations d'un aéronef et la dynamique des quaternions suivent des lois courbes et non rectilignes.

## 3 Vecteur d'État et Modélisation

### 3.1 Vecteur d'état (16 dimensions)

Pour décrire complètement le système, nous définissons le vecteur d'état suivant :

$$\mathbf{x} = \underbrace{[q_0, q_1, q_2, q_3]}_{\text{quaternion (4)}}, \underbrace{[p_x, p_y, p_z]}_{\text{position NED (3)}}, \underbrace{[v_x, v_y, v_z]}_{\text{vitesse NED (3)}}, \underbrace{[b_{\omega_x}, b_{\omega_y}, b_{\omega_z}]}_{\text{biais gyro (3)}}, \underbrace{[b_{a_x}, b_{a_y}, b_{a_z}]}_{\text{biais accel (3)}}^T \quad (1)$$

L'utilisation d'un **quaternion**  $\mathbf{q}$  permet de représenter l'orientation sans subir le "blocage de cardan" (gimbal lock) propre aux angles d'Euler. Nous intégrons également les **biais** des gyroscopes et des accéléromètres directement dans l'état. Ces capteurs sont en effet sensibles aux gradients thermiques de la stratosphère ; sans cette estimation continue, la dérive rendrait la navigation impossible après quelques minutes.

### 3.2 Modèle de prédition

À chaque pas de temps  $\Delta t$ , l'état est propagé selon les équations de mouvement discrétisées :

$$\mathbf{q}_{k+1} = \mathbf{q}_k + \frac{1}{2}\Omega(\boldsymbol{\omega}_{\text{mesuré}} - \mathbf{b}_\omega)\mathbf{q}_k \cdot \Delta t \quad (2)$$

$$\mathbf{p}_{k+1} = \mathbf{p}_k + \mathbf{v}_k \cdot \Delta t \quad (3)$$

$$\mathbf{v}_{k+1} = \mathbf{v}_k + [\mathbf{R}(\mathbf{q}) \cdot (\mathbf{a}_{\text{mesuré}} - \mathbf{b}_a) + \mathbf{g}] \cdot \Delta t \quad (4)$$

Où  $\mathbf{g} = [0, 0, 9.81]^T$  en repère NED (North-East-Down). La matrice de covariance  $\mathbf{P}$ , qui représente la confiance du filtre, est propagée via la Jacobienne  $\mathbf{F}$  du système :  $\mathbf{P}_{k+1} = \mathbf{F}\mathbf{P}_k\mathbf{F}^T + \mathbf{Q}$ , où  $\mathbf{Q}$  représente le bruit de processus.

### 3.3 Modèles de mesure (Correction)

Le filtre ajuste ses prédictions grâce à trois sources de correction :

- **Référence gravitaire (Accéléromètre)** : En vol stabilisé, l'accéléromètre mesure la gravité. Cela permet de corriger les erreurs de roulis et de tangage. *Condition de sécurité* : cette correction n'est appliquée que si la norme de l'accélération est proche de  $g$  ( $\pm 0.5 \text{ m/s}^2$ ), afin d'éviter de fausser l'attitude lors de manœuvres brusques.
- **Position et Vitesse GPS** : Fournissent une référence absolue de position et de vitesse à 10 Hz, empêchant la dérive à long terme de l'intégration inertie.
- **Cap GPS (Heading)** : Puisque le planeur n'a pas de magnétomètre, le cap (yaw) est déduit du vecteur vitesse horizontale :  $\psi = \arctan 2(v_y, v_x)$ . Cette correction n'est valide que si le planeur a une vitesse sol suffisante ( $> 2.5 \text{ m/s}$ ).

## 4 Stratégie d'Implémentation

### 4.1 Intégration Matérielle et Logicielle

La Figure 2 détaille le schéma de connexion entre les différents composants du système.

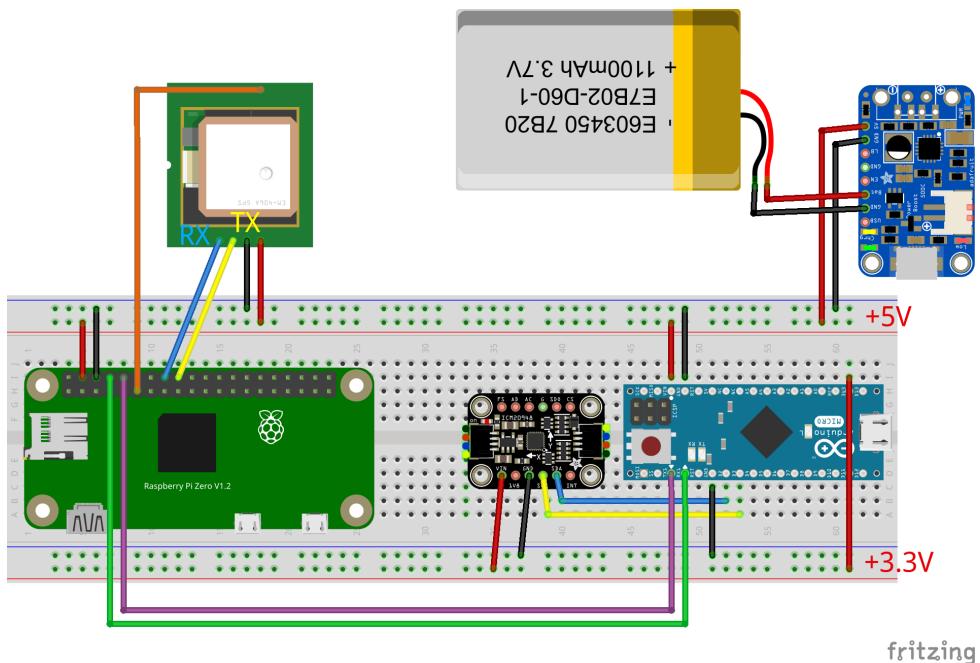


FIGURE 2 – Schéma de câblage des composants (Raspberry Pi, Arduino, GPS, IMU)

L'architecture de communication a été conçue pour optimiser les flux de données selon les contraintes de fréquence de chaque capteur :

- **Liaison Raspberry Pi ↔ Arduino** : Une connexion UART matérielle (Hardware UART) a été privilégiée. Ce choix est imposé par la fréquence d'échantillonnage élevée de l'IMU (100 Hz), nécessitant une transmission rapide, fiable et sans latence que seule l'UART native peut garantir.

- **Liaison Raspberry Pi ↔ GPS** : Le module GPS fonctionnant à une fréquence plus faible (10 Hz), une UART logicielle (Software UART) émulée sur les broches GPIO du Raspberry Pi est suffisante pour assurer la transmission des trames NMEA sans perte d'information.

L'intégralité du code source développé pour ce projet, ainsi que des tutoriels vidéo détaillant la procédure d'installation, sont disponibles sur le dépôt GitHub suivant :

[https://github.com/pierre9009/Planeur\\_code](https://github.com/pierre9009/Planeur_code)

## 4.2 Initialisation et Calibration

Avant le largage, une phase stationnaire de 30 secondes est nécessaire. Le filtre accumule environ 3000 échantillons pour moyenner les biais initiaux et stabiliser l'horizon. Un algorithme de détection de mouvement surveille les écarts-types ( $\sigma$ ) de l'IMU pour s'assurer que la calibration n'est pas perturbée par des vibrations extérieures.

## 4.3 Fusion adaptative selon la phase de vol

Le comportement du filtre s'adapte dynamiquement aux phases de la mission :

Phase	Dynamique	Stratégie de fusion
<b>Ascension</b>	Lente	Priorité au GPS et à la correction de l'accéléromètre pour faire converger les biais.
<b>Largage</b>	Très haute	Désactivation temporaire de certaines corrections GPS si le signal est instable ; priorité au maintien de l'assiette par l'IMU.
<b>Vol plané</b>	Modérée	Fusion complète. Activation du cap GPS dès que la vitesse sol est suffisante.

## 4.4 Gestion spécifique du Cap (Yaw)

Le cap est la variable la plus difficile à estimer sans magnétomètre, car la gravité ne donne aucune information sur le Nord.

- Tant que le planeur est suspendu ou avance trop lentement, le yaw est maintenu à une valeur fixe (verrouillage numérique) pour éviter qu'il ne dérive de manière aléatoire.
- Dès que la condition de vitesse ( $v_h > 2.5 \text{ m/s}$ ) est remplie, le filtre utilise la trajectoire GPS pour "recalibrer" l'orientation du nez de l'appareil.

## 5 Résultats et Analyse

Pour valider l'implémentation de l'EKF, nous avons développé un simulateur Python générant des données IMU et GPS synthétiques à partir de trajectoires connues (Vérité Terrain). Cela permet de comparer précisément l'estimation du filtre face à la réalité mathématique.

## 5.1 Protocole de Simulation

Deux scénarios principaux ont été testés :

- **Baseline (Ligne droite)** : Un vol rectiligne à vitesse constante sans vent. Ce test valide les équations de propagation basiques et l'intégration de la vitesse.
- **Realistic (Complexe)** : Un scénario complet incluant des virages, des variations d'altitude et une perte temporaire de signal GPS. Ce test vise à éprouver la robustesse du filtre lors des changements d'attitude (Roulis/Tangage).

## 5.2 Analyse des performances

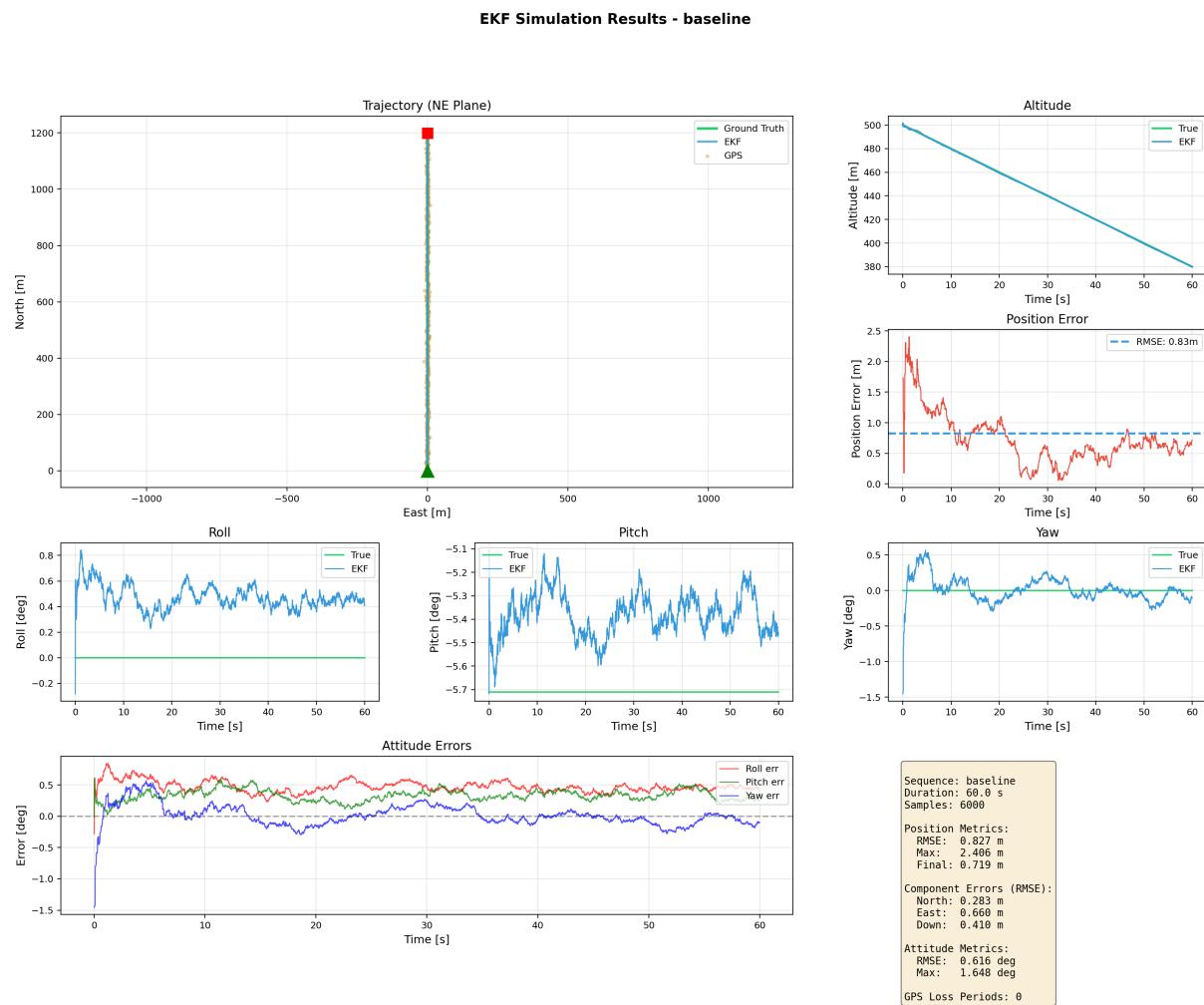


FIGURE 3 – Résultat de simulation : Scénario "Baseline" (Ligne droite). L'estimation (en bleu) suit parfaitement la vérité terrain (en vert).

Comme l'illustre la Figure 3, le filtre fonctionne de manière nominale sur une trajectoire simple. L'erreur de position reste bornée ( $\text{RMSE} < 1\text{m}$ ) et les biais des capteurs convergent correctement. Cela confirme que la structure matricielle de base de l'EKF et la gestion temporelle (prédiction/correction) sont fonctionnelles.

Cependant, le comportement se dégrade lors de manœuvres dynamiques, comme montré ci-dessous :

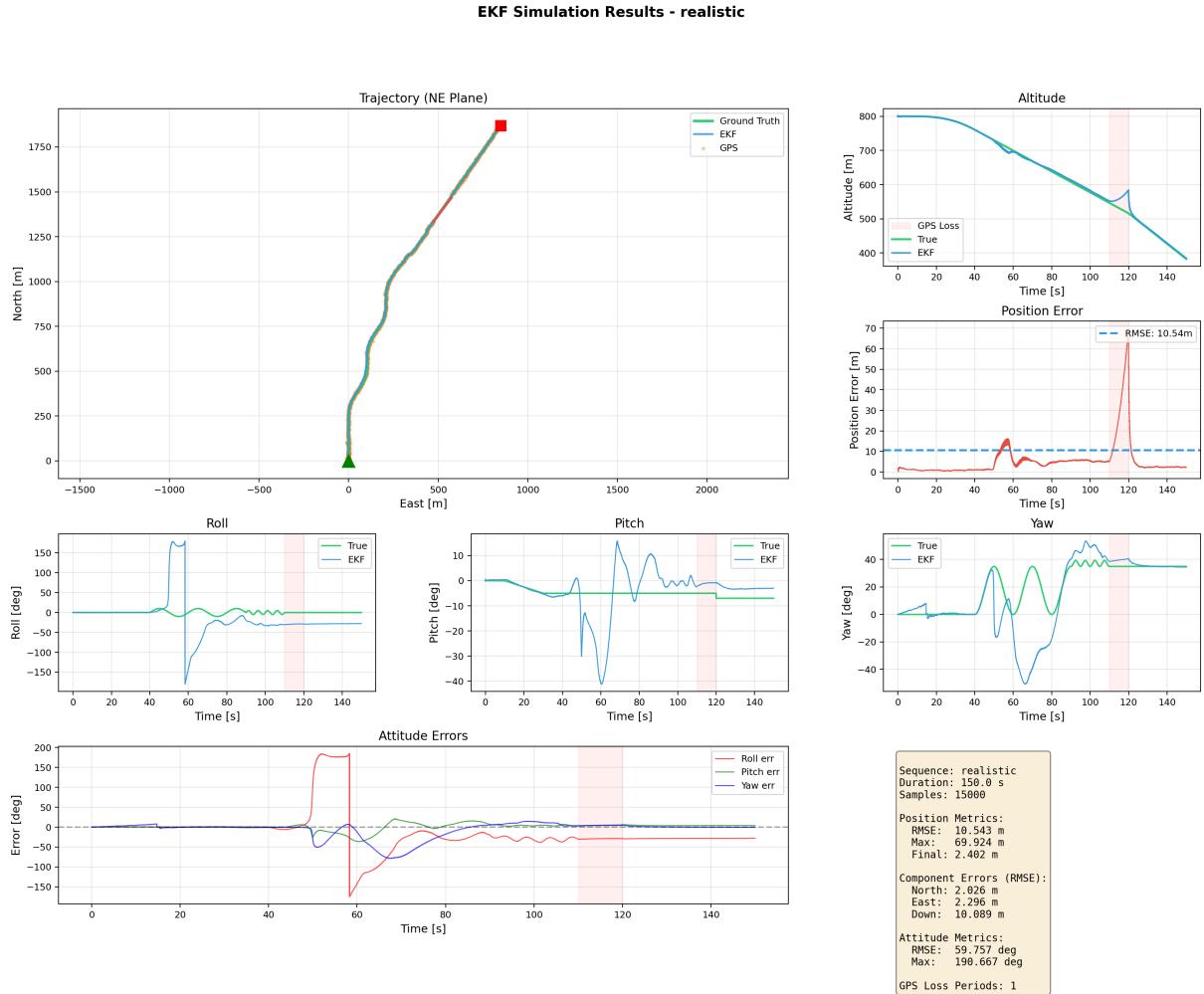


FIGURE 4 – Résultat de simulation : Scénario "Realistic". On note une divergence significative de l'attitude lors des virages ( $t=40$ s).

Sur la Figure 4, nous observons que :

- **En statique ou quasi-statique :** L'estimation des angles est correcte. Tant que le planeur ne subit pas d'accélérations latérales, l'horizon artificiel est stable.
- **En dynamique (virages) :** Une erreur critique apparaît lors de la mise en virage (autour de  $t=40$ s sur les graphes de Roll et Yaw). L'estimation de l'attitude se désynchronise de la réalité, ce qui entraîne une dérive rapide de la position (l'EKF projette le vecteur accélération dans la mauvaise direction).

Cette erreur suggère un problème de couplage entre l'estimation de l'attitude et la dynamique du véhicule. Il est probable qu'une incohérence persiste dans la gestion des repères lors des rotations (conflit entre le modèle de gravité et la force centrifuge mesurée), mais nous n'avons pas eu le temps d'isoler la cause exacte de ce bug avant la fin du projet.

### 5.3 Essais Réels

Compte tenu des instabilités détectées en simulation sur les trajectoires complexes, nous avons pris la décision de ne pas effectuer d'essais en vol réel. L'erreur d'estimation d'attitude en virage étant supérieure à la tolérance de sécurité, le risque de divergence de la navigation était trop élevé. Les tests se sont limités à des validations statiques au sol (IMU posée à plat), confirmant que l'acquisition des données et l'estimation de l'attitude au repos fonctionnent.

## 6 Conclusion et Perspectives

Ce projet a permis de mettre en place l'architecture complète d'un système de navigation inertielle assisté par GPS, depuis l'acquisition bas niveau (ICM-20948) jusqu'au filtrage optimal (EKF). Si le filtre est capable de fusionner les données et de maintenir une estimation cohérente en ligne droite, il reste vulnérable aux fortes dynamiques angulaires.

### 6.1 Conseils pour la suite du projet

Pour les futurs développeurs reprenant ce travail, nous recommandons de focaliser les efforts sur les points suivants :

1. **Rigueur absolue sur les conventions** : La majorité des erreurs en navigation inertielle proviennent de confusions de repères (NED vs ENU) ou de signes. Ne jamais supposer qu'un axe est positif sans vérifier.
2. **Attention à la Datasheet ICM-20948** : Un piège critique se situe **page 83 de la datasheet**. Le repère du capteur est défini en "Z-Up" (Z vers le haut), alors que les équations standard de l'aéronautique (NED) utilisent "Z-Down". Cette inversion doit être traitée avec une extrême vigilance dans le mapping des données brutes vers l'EKF.
3. **Priorité à la Simulation** : Il est impératif d'obtenir une estimation parfaite en simulation (erreur < 1°) sur des trajectoires complexes avant de tenter le moindre vol. Le simulateur est le seul endroit où l'on peut déboguer les matrices de rotation sans bruit parasite.

Le prochain objectif technique sera de corriger le modèle de prédiction lors des virages coordonnés, potentiellement en désactivant la correction de l'accéléromètre (qui mesure alors la force centrifuge et non la gravité) lorsque la vitesse angulaire est élevée.

Référence pour mieux comprendre l'EKF pour les prochains, attention y'a quelques erreurs de dérivées ou d'ordre de calcul mais le principal est là :

<https://medium.com/@opinoquintana/i-wrote-an-extended-kalman-filter-for-uav-attitude-estimation-from-scratch-in-rust-b8748ff33b12>

Bon courage !