



**HAL**  
open science

## Documentation interne V1

Imane Lemammer

► **To cite this version:**

| Imane Lemammer. Documentation interne V1. 2019. hal-02058906

**HAL Id: hal-02058906**

**<https://hal.science/hal-02058906>**

Preprint submitted on 6 Mar 2019

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Documentation interne V1

Lemammer Imane

## Table des matières

I- Introduction.....	3
II- Géométrie du C-arm et notations .....	4
1- Tableau de notations.....	4
2- Description de la géométrie du C-arm .....	4
III- Capteurs inertiels .....	6
1- Erreur et modèle des gyromètres .....	6
a- Erreurs inhérents aux gyromètres.....	6
b- Modèles de mesure des gyromètres.....	7
2- Erreur et modèle des accéléromètres :.....	8
3- Centrales à inertie utilisées .....	10
IV- Algorithmes d'estimation .....	11
1- Différentes représentation de l'orientation.....	11
a- Angles d'Euler et matrices de rotation .....	12
b- Quaternions.....	13
c- Passage d'une représentation d'orientation à l'autre.....	14
2- Algorithmes d'estimation à partir des mesures uniquement.....	17
a- Navigation à l'estime .....	17
b- Filtre complémentaire .....	18
3- Algorithmes d'estimation à partir des mesures et de modèles .....	20
a- Choix du filtre de Kalman étendu additif contraint.....	20
b- Principe du Filtre de Kalman étendu .....	21
c- Deux KF différents pour l'orientation et la position .....	21
d- Implémentation du filtre de Kalman.....	22
e- Initialisation.....	24
f- Echantillonnage .....	25
V- Calcul des paramètres de la matrice de projection .....	30
VI- Calcul de la matrice de projection.....	31
1- Idée générale .....	31
2- Détails .....	32
a- Changement de repère de navigation {w} au repère détecteur virtuel {d'}.....	32
b- Changement de repère du détecteur virtuel {d'} vers le repère source {s}.....	32

c- Projection du point de $\{w\}$ sur le plan détecteur $\{d\}$ .....	33
d- Changement de repère du repère source $\{s\}$ au repère détecteur $\{d\}$ .....	33
3- Transformation globale.....	33
VII- Propagation d'erreur .....	35
1- Formule globale de propagation d'Erreur.....	35
2 - Dérivée de la matrice de projection par rapport aux distances .....	35
a- Dérivée par rapport à $s_{dd}$ .....	35
b- Dérivée par rapport à $s_{id}$ .....	35
c- Dérivée par rapport à $s_x$ .....	36
d- Dérivée par rapport à $s_y$ .....	36
e- Dérivée par rapport à $s_x$ .....	36
f- Dérivée par rapport à $s_y$ .....	36
3- Dérivée de la matrice de projection par rapport aux angles. ....	36
a- Formules .....	36
b- Approximation petits angles .....	37
4- Influence des différents paramètres sur le projeté 2D : .....	37
a- Erreur selon $s_x$ et $s_y$ .....	38
b- Erreur selon $s_x$ et $s_y$ .....	39
c- Erreur selon $s_{wd}$ et $s_{dd}$ .....	40
d- Erreur selon $\varphi$ , $\theta$ et $\psi$ .....	41
VIII- Bras rotatif.....	43
1- Présentation du bras rotatif :.....	43
2- Stratégies de mouvement du bras rotatif .....	43
3- Caractérisation de la table tournante.....	45
4- Répétabilité de la table tournante .....	49
5- Petite discussion autour des pulsations.....	51
6- Simulateur.....	51
7- Conclusion : de la table tournante au C-arm.....	52
IX- Première estimation d'erreur .....	53
1- Des sorties de capteurs aux mesures.....	53
2- Réglage des filtres de Kalman.....	53
a- Matrice de covariance de mesures $R_a$ et $R_\omega$ .....	53
b- Matrice de covariance d'état $Q_a$ et $Q_\omega$ .....	55
3- Erreur d'estimation de pose.....	56
a- Erreurs d'estimation de pose pour une rotation continue.....	56

b- Remarque sur l'erreur d'estimation pour une rotation discontinue.....	57
X- Calibration :.....	58
1- Equations de calibrations.....	58
2- Calibration sans équipement.....	59
Signal de la calibration :.....	59
3- Test de l'algorithme.....	64
XI- Initialisation de algorithmes (caméra et IMU).....	65
Annexe 1 : Fiche des capteurs utilisés.....	66
1- NavChip ISNC02.....	66
2- Mti-1-8A7G6 (Xsens).....	67
3 - Yost-Lab 3space.....	68
4 – MPU-6050.....	69
5- Angular random walk (ARW) et Velocity random walk.....	70
Annexe 2 : Alternatives au Filtre de Kalman étendu et comparaison.....	71
Annexe 3 : Alternatives au Filtre de Kalman étendu et comparaison.....	72
Backward smoothing EKF (BSEKF).....	72
Observateur non linéaire [47-48-50-51-52].....	73
Filtre à particules [32 33 104].....	73
Méthodes QUEST (c'est le filtrage complémentaire) [32 33 104].....	73
Unscented Kalman filter (UKF) [32 33 104].....	73
Comparaison des différentes algorithmes pour la localisation d'un robot.....	73
Annexe 4 : Illustration des ordres d'échantillonnage.....	75
Méthode d'Euler (Ordre 1).....	75
Méthode Midpoint (Ordre 2).....	75
Méthode RK4 (Ordre 4).....	76

## **I- Introduction**

## II- Géométrie du C-arm et notations

### 1- Tableau de notations

Symbole	Sens
$a$	Scalaire
$\mathbf{a}$	Vecteur
$\mathbf{A}$	Matrice
$q$	Quaternion
$\dot{a}/\ddot{a}$	Dérivée temporelle d'ordre 1/d'ordre 2 de $a$
$\mathbf{a}^t/\mathbf{A}^t$	Transposée de $\mathbf{a}/\mathbf{A}$
$\text{Diag}(\mathbf{a})$	Matrice diagonale de diagonale le vecteur $\mathbf{a}$
$\mathbf{I}$	Matrice identité de la taille adéquate
$\{w\}$	Repère d'espace orthonormé direct
${}^w\mathbf{a}$	Vecteur $\mathbf{a}$ exprimé dans le repère $w$
$\mathbf{a}_k$	Vecteur $\mathbf{a}$ à l'échantillon $k$
${}^b_wq/{}^b_wR$	Quaternion/ matrice de rotation (définition en Seq. IV-1-b) de la rotation de $\{w\}$ à $\{b\}$ (a.k.a) rotation de $\{b\}$ par rapport à $\{w\}$

Tab. 1 : Tableau des notations générales utilisées dans le document

### 2- Description de la géométrie du C-arm

La géométrie proposée pour le C-arm est présentée dans la figure 1 et les notations utilisées sont présentées dans le Tableau 1.

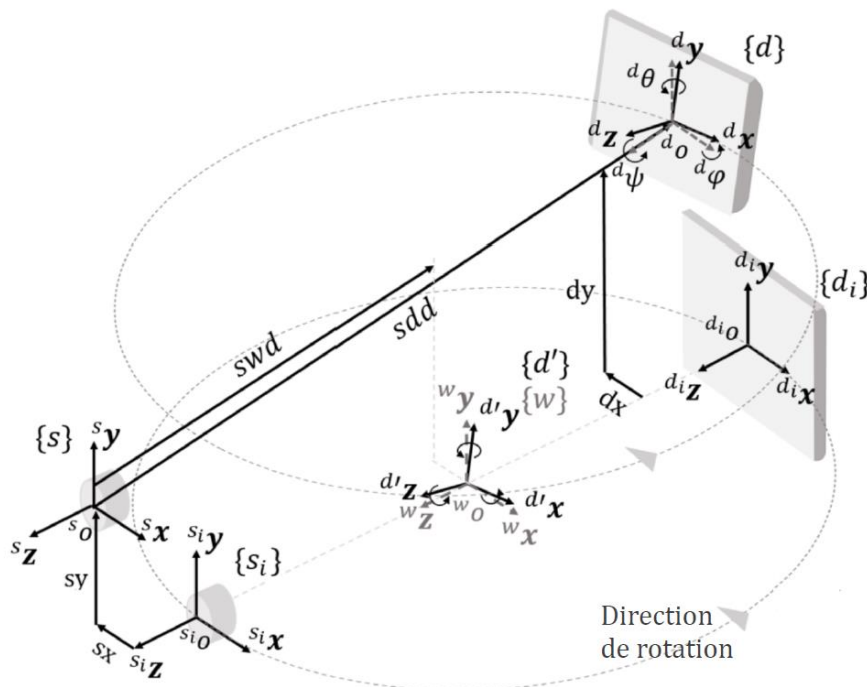


Fig. 1 : Géométrie du C-arm mobile définie par 4 repères orthonormés directs en comparaison avec une trajectoire de C-arm idéale

Dans ce document le C-arm est décrit par quatre repères orthonormés directs :  $\{w\}$ ,  $\{d\}$ ,  $\{d'\}$  et  $\{s\}$ . Ces repères interviendront dans les problèmes d'estimation de pose (orientation et position) du détecteur et de la source *Seq.IV*, de calcul des paramètres de projection *Seq.V* et de calcul de matrice de projection *Seq.VI*.

- **Le repère navigation  $\{w\}$**  de centre  ${}^w o$  et d'axes  $({}^w x, {}^w y, {}^w z)$  qui est le repère fixe autour duquel tourne le C et dans lequel la reconstruction tomographique est effectuée. **Remarque : comme nos durées d'acquisitions sont courtes (de l'ordre de 1min) le repère navigation est supposé fixe par rapport au repère terrestre.**
- **Le repère détecteur  $\{d\}$**  de centre  ${}^d o$  et d'axes  $({}^d x, {}^d y, {}^d z)$  qui est le repère lié au détecteur plan du C-arm
- **Le repère détecteur virtuel  $\{d'\}$**  de centre  ${}^w o$  et d'axes  $({}^{d'} x, {}^{d'} y, {}^{d'} z)$  est un repère de même orientation que  $\{d\}$  mais qui a la même origine que le repère navigation  $\{w\}$  (Ce repère interviendra uniquement lors du calcul de la matrice de projection du C-arm (*Seq.VI*)).
- **Le repère source  $\{s\}$**  de centre  ${}^s o$  et d'axes  $({}^s x, {}^s y, {}^s z)$  qui est le repère lié à la source rayons X du C-arm.

Les repère  $\{s_i\}$  et  $\{d_i\}$  représentent la source et le détecteur d'un C-arm « idéal », à savoir un C-arm dont la trajectoire est circulaire, isocentrique et reproductible. Ils sont illustrés dans la *Fig. 1* à titre de comparaison.

Pour une CBCT, des projections 2D sont acquises durant une rotation continue de l'arceau. Nous étudions dans ce document rotation autour d'un seul axe  ${}^w y$ . En raison des jeux et des flexions mécaniques, les trajectoires du détecteur et de la source d'écartent sensiblement de la trajectoire circulaire idéale représentée par  $\{s_i\}$  et  $\{d_i\}$  suivant 3 translations et 3 rotations. Ces mouvements ne sont pas reproductibles.

Les jeux angulaires maximaux observés pour la source rayon X d'un C-arm classique sont supposés petits. On suppose, donc, que la **source à rayons X est isotopique**. Par conséquent, l'inclinaison de la source peut être négligée lors du calcul de la matrice de projection (*Seq.VI*). Toutefois, elle intervient dans nos estimateurs de position de la source (*Seq.IV*) et dans le calcul des paramètres de la matrice de projection (*Seq.V*). Dans ce document, le **bâti du C-arm est également supposé fixe** par rapport au repère navigation  $\{w\}$ .

On nomme **piercing point**  $pp$  la projection de l'origine du repère source  $\{s\}$  sur le plan du détecteur. Il est de coordonnées  $[sx - dx \quad sy - dy]^t$

### III- Capteurs inertiels

Dans cette partie nous posons :

- $\{b\}$  le repère du solide qui est défini comme le repère lié au capteur et centré à l'origine de l'accéléromètre, les axes sont alignés au boîtier. Ce repère représente soit le repère détecteur  $\{d\}$  soit le repère source du C-arm  $\{s\}$ .
- $\{w\}$  le repère de navigation fixe défini dans *Seq. II-2*.

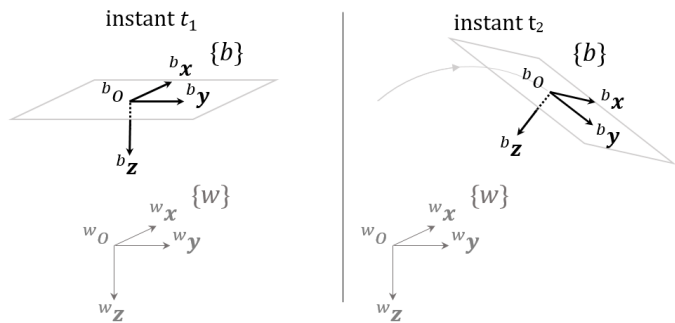


Fig. 2 : Repères pour les IMUs

La même étude peut être conduite pour le détecteur et pour la source. On va, donc, se limiter à l'étude de la localisation d'un seul des deux que l'on nomme « solide ».

#### 1- Erreur et modèle des gyromètres

##### a- Erreurs inhérents aux gyromètres

Les mesures de vitesse angulaire d'un gyromètre sont entachées d'erreurs qu'on va essayer de caractériser rapidement dans cette partie. On note E le signal d'entrée du capteur et S son signal de sortie.

- **Facteur d'échelle (Gain)** : Le rapport d'échelle est le facteur de proportionnalité entre S et E. Le facteur d'échelle est le plus souvent constant.
- **Biais** : Même si  $E = 0$  il se peut que  $S \neq 0$ . La valeur de S, dans ces conditions, est la valeur du Biais.
- **Bruit** : Empiriquement, nous remarquons que le bruit qui domine nos gyromètres est un bruit gaussien centré (*Seq. IX-2*) (il en est de même pour les accéléromètres). Vu nos durée d'intégrations courtes, les autres modèles de bruit sont négligeable comme illustré sur la Fig.3. **Ref**

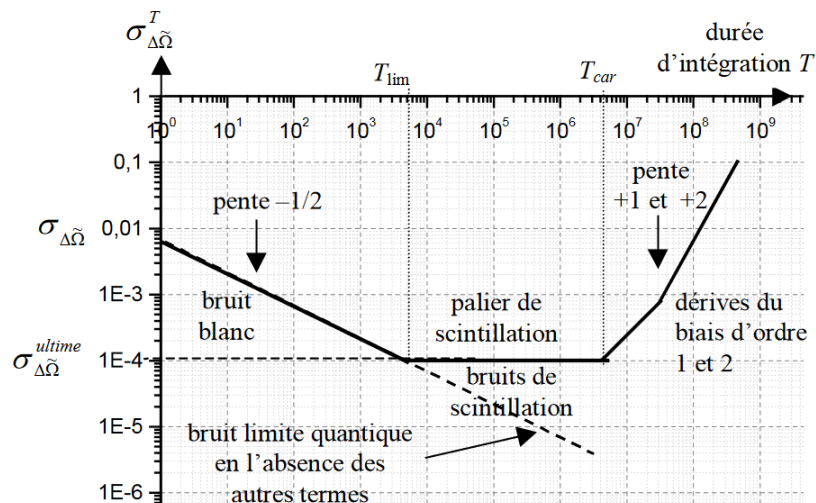


Fig. 3 : Type de bruit en fonction de la durée d'intégration pour un gyromètre.

- **Dynamique** : Plage de valeur de la grandeur d'entrée E qui donne une valeur correcte du signal de sortie S.
- **Bande passante** : Tous les capteurs agissent comme des filtres passe-bas avec une fréquence de coupure plus ou moins élevée. Si la vitesse de rotation oscille entre deux valeurs avec une fréquence très rapide, l'appareil ne distinguera pas ces variations (dans le meilleur des cas on obtient une valeur moyenne intégrée sur le temps de réponse de l'appareil.). Il suffit de choisir un appareil avec une bande passante adéquate.

Nous avons choisi des capteurs adéquats. Nous pouvons considérer que **les erreurs dus à la dynamique ou à la bande passante sont négligeables.**

- En pratique tous les gyromètres sont **sensibles aux accélérations** (à cause de l'asymétrie du design mécanique/inexactitudes d'usinage, la sévérité de la sensibilité varie d'un design à l'autre). Globalement, les IMUs récentes sont robustes vis-à-vis de la sensibilité en  $g^2$  **ref**. En ce qui concerne les IMUs dont on dispose, cette source d'erreur n'est pas spécifiée dans la datasheet donc **négligeable**.
- Certains de nos gyromètres sont sensibles aux accélérations en  $g$  **ref**. Cependant, Au vu du mouvement du C-arm qui est essentiellement une rotation les accélérations dues à des translations du système sont négligeables donc le terme en de la sensibilité en  $g$  n'interviendra pas dans notre modèle de mesure final.
- La stabilité du biais et l'une des sources d'erreur les moins importantes pour les gyromètres dont on dispose (voir Annexe 1 et Datasheet de capteurs **ref ref ref** et **ref**). **Et au vu de nos durées d'intégration (~1min) elle est négligeable.**
- Sensibilité aux variations de température : Les variations de température introduisent souvent un changement de gain et de biais des capteurs. Cependant, certains de nos capteurs sont compensés et température (Seq.III-3). Ce n'est pas le cas pour les IMUs « low-cost » comme le MPU6050. Néanmoins, durant nos acquisitions nous observons un écart type de **température inférieure à 0.01°**. Reporté sur la mesure cela entraîne une **erreur type négligeable de 0.0004° pour le gyromètre et de 0.0002m/s<sup>2</sup> pour l'accéléromètre**. Nous pouvons, donc, considérer qu'on travaille à température constante après un étalonnage. Par conséquent  $S_{\omega,k} = S_{\omega}$  et  $\beta_{\omega,k} = \beta_{\omega}$  sont constants.

## b- Modèles de mesure des gyromètres

Un modèle complet de la mesure d'un gyromètre d'écrit pour un échantillon  $k$

$$y_{\omega,k} = M_{\omega} \left( S_{\omega,k}(\theta) \times {}^b_w \omega_k + C_{\omega,a} {}^w a_{tr,k} + G_s \omega_{gen} - \beta_{\omega,k}(\theta) \right) + {}^b b_{\omega} + {}^b n_{\omega,k}$$

$y_{\omega,k}$  : Mesure du gyromètre (en rad/s)

${}^b \omega_k$  : Vitesse angulaire mesurée exprimée dans le repère du solide (en rad/s)

${}^b b_{\omega}$  : Biais du gyromètre (en rad/s)

${}^b n_{\omega,k}$  : Bruit du gyromètre (en rad/s)

$M_{\omega}$  : Matrice de gain et d'influence entre axes du gyromètre (sans unité)



$$M_{\omega} = \begin{pmatrix} gain_x & misalignement_{xy} & misalignement_{xz} \\ misalignement_{xy} & gain_y & misalignement_{yz} \\ misalignement_{xz} & misalignement_{yz} & gain_z \end{pmatrix}$$

$\theta$  : Température (Sans unité)

$S_{\omega,k}(\theta)$  : Matrice diagonale de la sensibilité du gyromètre en fonction de la température (Sans unité)

$\beta_{\omega,k}(\theta)$  : Vecteur des Biais dus aux variations de la température (en rad/s)

$a_{tr,c}$  : Accélération issue de la translation du solide (en  $m/s^2$ )

$C_{\omega,a}$  : Matrice diagonale de la sensibilité du gyromètre aux accélérations linéaires

$\omega_{gen}$  : Terme en  $g^2$ , vibrations (SI)

$G_s$  : Matrice diagonale de la sensibilité du gyromètre aux vibrations en  $g^2$

Au vu des simplifications présentée en **(III-1-a)**, nous posons  $G_s = 0_{3 \times 3}$  et  ${}^w a_{tr,k} = 0_{3 \times 1}$ . De même  $S_{\omega,k}(\theta)$  et  $\beta_{\omega,k}(\theta)$  sont constants et nous pouvons écrire  $S_{\omega,k}(\theta) = S_{\omega}$  et  $\beta_{\omega,k}(\theta) = \beta_{\omega}$

Nous obtenons :

$$\begin{aligned} y_{\omega,k} &= M_{\omega}(S_{\omega} \times {}^b \omega_k - \beta_{\omega}) + {}^b b_{\omega} + {}^b n_{\omega,k} \\ y_{\omega,k} &= M_{\omega} S_{\omega} {}^b \omega_k + (-M_{\omega} \beta_{\omega} + {}^b b_{\omega}) + {}^b n_{\omega,k} \end{aligned}$$

d'où :

$$y_{\omega,k} = M'_{\omega} {}^b \omega_k + {}^b b'_{\omega} + {}^b n_{\omega,k} \quad (1)$$

$M'_{\omega} = M_{\omega} S_{\omega}$  : Matrice de gain et de désalignements axiaux du gyromètre

${}^b \omega_k$  : Vitesse angulaire mesurée exprimé dans {b}(en rad/s)

${}^b b'_{\omega} = (-M_{\omega} \beta_{\omega} + {}^b b_{\omega})$ : Biais du gyromètre(en rad/s)

${}^b n_{\omega,k}$  : Bruit du gyromètre(en rad/s)

## 2- Erreur et modèle des accéléromètres :

Un accéléromètre est globalement sujet aux même erreurs que le gyromètre (présentée en Seq.III-1-a) (mis appart les sensibilités en g et en  $g^2$ ). Nous allons également pouvoir faire les mêmes simplifications de modèle que pour le gyromètre.

Un accéléromètre peut être vu comme une masse en suspension grâce à un bras de levier. La gravité accélère la masse mais cause aussi la flexion du bras de levier. La sortie d'un accéléromètre peut être donnée par l'équation (exprimé dans le repère du solide, voir partie 1) :

$$a_{masse} = \frac{1}{m} (F - F_g)$$

$a_{masse}$ : acceleration de la masse (en  $m/s^2$ )

$m$  : masse de la masse en suspension (en g)

$F$  : Ensemble des forces appliquées à la masse (en N), gravité comprise

$F_g$  : Force appliquée par la gravité (en N)

Nous allons calculer les accélérations de notre solide par rapport au repère initial à partir des accélérations mesurées par le capteur. Considérons que nous disposons d'un accéléromètre idéal qui mesure les accélérations  $a_{mesuree}$  selon les 3 axes  ${}^b x$ ,  ${}^b y$  et  ${}^b z$  du Repère Solide {b}.

Un accéléromètre idéal mesure une somme de l'accélération relative et des accélérations issues de la contribution des forces normales appliquées à l'accéléromètre. Cela donne en équation :

$${}^w\mathbf{a}_{mesuree} = {}^w\mathbf{a} + {}^w\mathbf{a}_{gravite} + {}^w\mathbf{a}_{coriolis}$$

${}^w\mathbf{a}_{mesuree}$  : Accélération mesurée par un accéléromètre exprimée dans  $\{w\}$ (en  $m/s^2$ )

${}^w\mathbf{a}$  : Accélération relative du corps exprimée dans  $\{w\}$ (en  $m/s^2$ )

${}^w\mathbf{a}_{gravite}$  : Accélération due à la gravité terrestre exprimée dans  $\{w\}$ (en  $m/s^2$ )  
(approximativement  $9,81 m/s^2$ )

${}^w\mathbf{a}_{coriolis}$  : Accélération de coriolis exprimée dans  $\{w\}$

Une simple application permet de montrer que  $|{}^w\mathbf{a}_{coriolis}| = |-2\boldsymbol{\omega}_{n/terre} \wedge {}^R\mathbf{v}_M| \sim 10^{-4}m/s^2$  avec  $\omega_{n/terre}$  vitesse angulaire de la rotation de la terre et  ${}^R\mathbf{v}_M$  une valeur type de vitesse de notre mobile. Cette valeur est inférieure à la valeur de notre bruit d'accélération Seq.IX-2. On considère que  ${}^w\mathbf{a}_{coriolis} = 0$ . L'équation devient :

$${}^b\mathbf{a}_{mesurée,k} = {}^b\mathbf{R}_k({}^w\mathbf{a}_k - {}^w\mathbf{g})$$

${}^b\mathbf{R}_k$  : la matrice de rotation de  $\{b\}$  par rapport à  $\{w\}$

${}^w\mathbf{g}$  : le vecteur de gravité terrestre exprimé en  $\{w\}$ en  $m/s^2$  ici orienté selon  ${}^w\mathbf{y}$

Le modèle de mesure complet de l'accéléromètre s'écrit :

$$\mathbf{y}_{a,k} = \mathbf{M}_a \left( \mathbf{S}_{a,k}(\theta) \times {}^b\mathbf{R}_k({}^w\mathbf{a}_k - {}^w\mathbf{g}) - \boldsymbol{\beta}_{a,k}(\theta) \right) + {}^b\mathbf{b}_a + {}^b\mathbf{n}_{a,k}$$

$\mathbf{y}_{a,k}$ : mesure de l'accéléromètre (en  $m/s^2$ )

$\mathbf{M}_a$  : Matrice de gain de désalignements axiaux de l'accéléromètre

$\mathbf{S}_{a,k}(\theta)$  : Matrice diagonale de la sensibilité de l'accéléromètre à la température

${}^b\mathbf{R}_k$  : Matrice de rotation de  $\{w\}$  à  $\{b\}$

${}^w\mathbf{a}_k$  : Accélération par rapport au repère monde (en  $m/s^2$ )

${}^w\mathbf{g}$  : vecteur de gravité (en  $m/s^2$ )

$\boldsymbol{\beta}_{a,k}(\theta)$  : Vecteur des Biais dûs aux variations de la température (en  $m/s^2$ )

${}^b\mathbf{b}_{a,k}$  : biais de l'accéléromètre exprimé dans  $\{b\}$ (en  $m/s^2$ )

${}^b\mathbf{n}_{a,k}$  : bruit de l'accéléromètre exprimé dans  $\{b\}$ (en  $m/s^2$ )

En raisonnant comme pour le gyromètre nous obtenons :

$$\mathbf{y}_{a,k} = \mathbf{M}'_a {}^b\mathbf{R}_k({}^w\mathbf{a}_k - {}^w\mathbf{g}) + {}^b\mathbf{b}'_a + {}^b\mathbf{n}_{a,k} \quad (2)$$

$\mathbf{M}'_a = \mathbf{M}_a\mathbf{S}_a$  : Matrice de gain et de désalignement axial de l'accéléromètre

${}^b\mathbf{R}_k$  : Matrice de rotation de  $\{w\}$  à  $\{b\}$  à l'instant  $k$

${}^w\mathbf{a}_k$ : acceleration par rapport au repère monde (en  $m/s^2$ )

${}^w\mathbf{g}$ : vecteur de gravité (en  $m/s^2$ )

${}^b\mathbf{b}'_a = (-\mathbf{M}_a\boldsymbol{\beta}_a + {}^b\mathbf{b}_a)$ : biais de l'accéléromètre (en  $m/s^2$ )

${}^b\mathbf{n}_{a,k}$  :  $(3 \times 1)$  bruit de l'accéléromètre (en  $m/s^2$ )

### 3- Centrales à inertie utilisées

Nous disposons de 4 types de centrales à inertie. Deux types sont déjà mis en place sur notre plateforme expérimentale (le MPU6050 et le ISNC02). Plus d'informations, ainsi que les schémas blocs sont présentées dans **l'Annexe 1 « fiche capteurs »**.

- **MPU6050 de TDK-Invensense [ref](#)** : Centrale inertielle peu couteuse, nous renvoie les données d'un thermomètre, d'un gyromètre et d'un accéléromètre 3 axes filtrées par un filtre passe bas paramétrable. Le composant doit être calibré manuellement, et est sujet à des non linéarités mais offre un large choix de fréquence et de plages de mesures.
- **NavChip ISNC02 de Thales-Intersens [ref](#)** : Centrale inertielle haut de gamme, calibrée en usine avec compensation des désalignements des axes, et en température. Elle nous renvoie des données formatées :  $\Delta\theta$  et  $\Delta V$  qui sont des valeurs intégrées de la vitesse angulaire et de l'accélération respectivement, analogues à ces dernières mais pour lesquelles les non-linéarités ont été filtrées. Notons que les mesures renvoyées sont issues de différents capteurs de constructeurs différents intégrés à la carte du NavChip et fusionnés dans un EKF. Ce capteur se présente comme une boîte noire et nous renvoie des données à une fréquence de 200Hz avec des plages de mesures fixes.
- **Mti-1-8A7G6 de Xsens technologies [ref](#)** : comprend un gyromètre 3 axes, un accéléromètre 3 axes et un magnétomètre 3 axes. Comme pour le ISNC02, renvoie des données intégrées de la vitesse angulaire et de l'accélération. Ce composant est **pré-calibré** et est compensé en température et en désalignement des axes. L'acquisition se fait à 800Hz mais la fréquence d'échantillonnage de sortie est de 100 Hz max. Cette IMU est équipée d'un filtre passe bas de fréquence de coupure 255Hz. Comprends un bloc fusion qui permet, à partir des données des différents capteurs d'obtenir l'information d'orientation.
- **Yost-Lab 3space de YOST labs [ref](#)** : C'est le capteur présentant la plus grande flexibilité en ce qui concerne le choix de sortie à savoir :
  - Mesures brutes du gyromètre et de l'accéléromètre
  - Mesures « normalisées » des capteurs (représentatifs des grandeurs physiques mesurées)
  - Mesure de température
  - Orientation : Peut être renvoyée en quaternions absolus ou relatifs, matrice de rotation ou angle d'Euler
  - $\Delta q$  rotation incrémentale sur la période d'intégration analogue à une vitesse angulaire mais une partie de l'intégration est effectuée pour filtrer les non linéarités  $\Delta q$ . Permet de réduire le bruit
  - $\Delta V$  : intégrale de l'accélération sur une période de mesure analogue à une accélération mais une partie de l'intégration est effectuée pour filtrer les non linéarités  $\Delta V$ . Permet de réduire le bruit

Ce composant est calibré en usine : Gain, biais, désalignement des axes pour le gyromètre et l'accéléromètre. (Activable ou non)

## IV- Algorithmes d'estimation

### 1- Différentes représentation de l'orientation

Plusieurs outils ont été mis en place pour représenter l'orientation d'une base vectorielle par rapport à une autre. Il est légitime de se poser la question : **Avec quel objet mécanique va t'ont décrire nos rotations ?**

Les représentations principales sont :

- **Angles d'Euler** [ref] et les **angles de Rodrigues** [ref] qui utilisent le minimum nécessaire de trois paramètres pour décrire une rotation. Ces représentations peuvent sembler être une bonne idée pour limiter la redondance des paramètres. Cependant, elles sont évitées pour l'estimation globale de l'attitude et sont surtout utilisées pour définir les erreurs d'attitude locales en raison de leurs singularités. De plus, les angles d'Euler ne sont pas un paramétrage continu et souffrent du blocage du cardan [ref]. Les vecteurs Rodrigues utilisent des fonctions périodiques. Toutefois, les valeurs peuvent dériver quand la fonction cosinus du dénominateur passe par zéro.

Singularités des angles d'Euler : Le système d'angle est en singularité de représentation lorsque le deuxième angle est tel que les deux axes de rotation extrêmes sont alignés. Deux conséquences :

- Dédire des angles d'Euler à partir d'une matrice de rotation ne peut se faire sans ambiguïté
- Les dérivées angulaires ne peuvent plus être calculées à partir du vecteur de rotation (division par 0)

Ces singularités obligent des algorithmes numériques à l'approche des angles critiques pour empêcher la perte d'un degré de liberté (erreur d'arrondi volontaire) et conduisent à des comportements incohérents à l'approche des singularités (rotations parasites)...

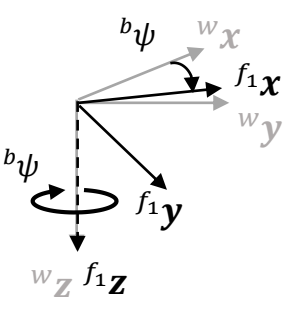
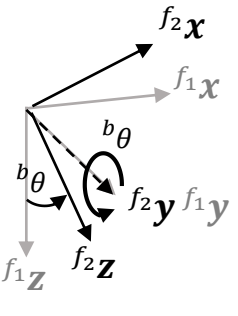
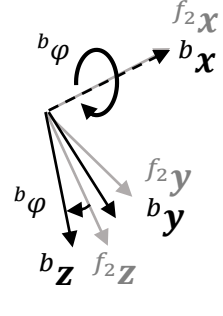
- Les **matrices de rotation** ne conviennent pas en raison du grand nombre de paramètres qui doivent être ajustés de manière cohérente (9 paramètres, 6 contraintes de l'exigence d'orthogonalité). De plus, l'orthogonalité des matrices n'est pas triviale à garantir.
- Les **représentations axis-angle** ont un seul paramètre redondant. Comme pour les angles d'Euler, les couples axis-angle souffrent de problèmes périodicité.
- Les **quaternions** sont la représentation d'orientation les plus utilisés dans les problèmes d'estimation d'orientation. En effet, leurs équations cinématiques sont linéaires, dénuées de singularités et des rotations successives sont opérées par simple multiplication quaternions. De plus, les quaternions sont construits sur la base de fonctions périodiques et sont compacts et plus numériquement stables que les matrices de rotation. De plus, multiplier un vecteur par un quaternion coûte encore moins cher que de passer par une multiplication vectorielle-matricielle complète.

Dans la suite nous allons essentiellement exploiter 3 représentations de rotations : essentiellement des **quaternions dans les algorithmes d'estimation** pour les qualités présentées ci-haut, les **angles d'Euler pour les affichages**, car ils sont plus facilement interprétables ainsi que comme paramètre de la matrice de projection. Nous allons également exploiter des **matrices de rotation lors des passages d'un repère à l'autre** et lors du calcul de la matrice de rotation. Dans la suite, nous allons développer ces 3 représentations ainsi que les algorithmes de passage de l'un à l'autre. Ces représentations sont interchangeables.

### a- Angles d'Euler et matrices de rotation

L'orientation du repère solide {b} par rapport au repère de navigation {w} peut être décrite par trois rotations directes (dans le sens inverse des aiguilles d'une montre) d'angles dits d'**Euler** autour de trois axes. Les trois angles sont définis dans Tab.2 (avec introduction de repères intermédiaires) et sont notés  ${}^b\varphi$ ,  ${}^b\theta$  et  ${}^b\psi$  selon les trois directions spatiales x-y-z. Les angles  ${}^b\varphi$ ,  ${}^b\theta$  et  ${}^b\psi$  sont appelés lacet, tangage et roulis respectivement.

Nous utilisons la convention (z-y-x) pour la définition des axes de rotation [ref]. **Cette dernière est la plus utilisée dans les problèmes de navigation et de localisation.** Sous la convention choisie, la matrice de rotation du repère solide {b} par rapport au repère navigation {w} s'écrit :

		
{w} → {f <sub>1</sub> }	{f <sub>1</sub> } → {f <sub>2</sub> }	{f <sub>2</sub> } → {b}
${}_{w}^{f_1}R({}^b\psi)$ $= \begin{bmatrix} \cos({}^b\psi) & \sin({}^b\psi) & 0 \\ -\sin({}^b\psi) & \cos({}^b\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}$	${}_{f_1}^{f_2}R({}^b\theta)$ $= \begin{bmatrix} \cos({}^b\theta) & 0 & -\sin({}^b\theta) \\ 0 & 1 & 0 \\ \sin({}^b\theta) & 0 & \cos({}^b\theta) \end{bmatrix}$	${}_{f_2}^bR({}^b\varphi)$ $= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos({}^b\varphi) & \sin({}^b\varphi) \\ 0 & -\sin({}^b\varphi) & \cos({}^b\varphi) \end{bmatrix}$

Tab.2 Détail du calcul de la matrice de rotation de {w} vers {b}

$${}_{w}^bR({}^b\varphi, {}^b\theta, {}^b\psi) = {}_{f_2}^bR({}^b\varphi) {}_{f_1}^{f_2}R({}^b\theta) {}_{w}^{f_1}R({}^b\psi)$$

avec {f<sub>1</sub>} et {f<sub>2</sub>} les repères intermédiaires définis dans le Tab. 2.

Après multiplication des 3 matrices on obtient :

$${}_{w}^bR({}^b\varphi, {}^b\theta, {}^b\psi) = \begin{bmatrix} c({}^b\theta)c({}^b\psi) & c({}^b\theta)s({}^b\psi) & -s({}^b\theta) \\ s({}^b\varphi)s({}^b\theta)c({}^b\psi) - c({}^b\varphi)s({}^b\psi) & s({}^b\varphi)s({}^b\theta)s({}^b\psi) + c({}^b\varphi)c({}^b\psi) & s({}^b\varphi)c({}^b\theta) \\ c({}^b\varphi)s({}^b\theta)c({}^b\psi) + s({}^b\varphi)s({}^b\psi) & c({}^b\varphi)s({}^b\theta)s({}^b\psi) - s({}^b\varphi)c({}^b\psi) & c({}^b\varphi)c({}^b\theta) \end{bmatrix} \quad (3)$$

c est le symbole pour cosinus et s est le symbole pour sinus

## b- Quaternions

Comme dit précédemment, les quaternions sont largement utilisées dans les algorithmes d'estimation d'orientation [ref\[8, 13\]](#). Ils sont plus simples à composer et numériquement plus stables que les autres représentations d'orientation. Un quaternion  $q$  est noté :

$$q = [q_0 \ q_1 \ q_2 \ q_3]^t$$

ici  $q_0$  est le composant scalaire du quaternion et  $\mathbf{q}_v = [q_1 \ q_2 \ q_3]^t$  est le composant vectoriel du quaternion. Les rotations sont représentées par des quaternions unitaires (quaternions de norme 1). Nous notons le quaternion de la rotation du repère solide {b} par rapport au repère navigation {w}  ${}^b_w q$ . Ce quaternion correspond à la matrice de rotation  ${}^b_w \mathbf{R}$ . On écrit dans la suite de cette sous partie  $q = {}^b_w q$ .

La rotation inverse : de {b} vers {w} est représentée par le quaternion conjugué  $q^*$  défini comme :

$$q^* = [q_0 \ -q_1 \ -q_2 \ -q_3]^t$$

La composition des rotations dans l'espace à trois dimensions revient à la multiplication de leurs représentations en quaternions (Cette multiplication est non commutative). Par exemple,  ${}^d_i q = {}^d_w q {}^w_i q$  représente la rotation de {i} vers {w} suivie de la rotation de {w} vers {d}. La multiplication des quaternions est calculée comme :

$$qp = \begin{bmatrix} p_0 q_0 - p_1 q_1 - p_2 q_2 - p_3 q_3 \\ p_0 q_1 + p_1 q_0 + p_2 q_3 - p_3 q_2 \\ p_0 q_2 - p_1 q_3 + p_2 q_0 + p_3 q_1 \\ p_0 q_3 + p_1 q_2 - p_2 q_1 + p_3 q_0 \end{bmatrix}$$

La dérivée temporelle d'un quaternion s'écrit :

$$\dot{q} = \frac{1}{2} \boldsymbol{\Omega} q$$

où  $\boldsymbol{\Omega}$  matrice de rotation relative :

$$\boldsymbol{\Omega} = \begin{bmatrix} 0 & \boldsymbol{\omega}^t \\ -\boldsymbol{\omega} & -\boldsymbol{\omega}^\times \end{bmatrix}$$

et  $\boldsymbol{\omega} = [\omega_x \ \omega_y \ \omega_z]^t$  est la vitesse angulaire du repère {b} par rapport au repère {w}. Et où  $\boldsymbol{\omega}^\times$  la matrice anti symétrique de la vitesse angulaire :

$$\boldsymbol{\omega}^\times = \begin{bmatrix} 0 & \omega_z & -\omega_y \\ -\omega_z & 0 & \omega_x \\ \omega_y & -\omega_x & 0 \end{bmatrix}$$

Nous avons besoin d'écrire la dérivée temporelle dans le domaine discret. L'équation de dérivation à temps discret s'écrit pour un temps  $k\Delta t$  ou  $k$  l'échantillon et  $\Delta t$  la période d'échantillonnage :

$$q_{k+1} = \exp(\boldsymbol{\Omega}_k \Delta t) q_k$$

Nous allons considérer que nous travaillons avec des fréquences d'échantillonnage suffisamment grandes ( $\geq 100\text{Hz}$ ). Les quantités  $\boldsymbol{\Omega}_k \Delta t$  peuvent alors être considérée comme très petites entre 2 échantillons consécutifs. Nous pouvons approximer  $\exp(\boldsymbol{\Omega}_k \Delta t)$  en utilisant la série

de Taylor correspondante. Pour l'instant, nous admettons que nous allons utiliser l'ordre 2 de la série de Taylor pour approximer notre exponentielle. La question de l'échantillonnage va être développée dans plus de détails dans *Seq.IV-3-f*. Nous obtenons :

$$q_{k+1} = \left( I + \frac{1}{2} \Omega_k \Delta t - \frac{1}{2!} \left( \frac{1}{2} \Omega_k \Delta t^2 \right) \right) q_k \quad (4)$$

### c- Passage d'une représentation d'orientation à l'autre

Adapté de : [https://en.wikipedia.org/wiki/Conversion\\_between\\_quaternions\\_and\\_Euler\\_angles](https://en.wikipedia.org/wiki/Conversion_between_quaternions_and_Euler_angles) pour convenir aux conventions choisies.

Comme nous manipulons 3 représentations différentes d'orientation, il est essentiel de développer les formules de passage d'une représentation à l'autre. Au vu de la multiplicité des conventions de matrices de rotations et de quaternions, les formules de passages ont été ré implémentées afin de convenir aux choix faits. Au vu des singularités et de la non unicité de représentation des angles d'Euler et des matrices de rotations, le passage d'une représentation à l'autre de l'orientation n'est pas si évident. Nous utilisons principalement les passages de **quaternions à angles d'Euler** – **d'angles d'Euler à quaternions** - de **quaternions à matrice de rotation** – **d'angles d'Euler à matrice de rotation**.

- **Quaternions à angle d'Euler**

On applique la formule

$$\begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} \arctan \frac{2(q_0 q_1 + q_2 q_3)}{1 - 2(q_1^2 + q_2^2)} \\ \arcsin(2(q_0 q_2 - q_3 q_1)) \\ \arctan \frac{2(q_0 q_3 + q_1 q_2)}{1 - 2(q_2^2 + q_3^2)} \end{bmatrix}$$

Comme les fonction arctan and arcsin en Matlab (comme en C++) renvoient des valeurs entre  $-\pi/2$  et  $\pi/2$ , on utilise la fonction « atan2 » pour obtenir l'ensemble des orientations possibles.

```
function [phi,teta,psi] = from_quat_to_eul(q)
    q_0 = q(1);    %q.w
    q_1 = q(2);    %q.x
    q_2 = q(3);    %q.y
    q_3 = q(4);    %q.z

    % phi(roll)
    sinr = 2.0 * (q_0 * q_1 + q_2 * q_3);
    cosr = 1.0 - 2.0 * (q_1 * q_1 + q_2 * q_2);
    phi = atan2(sinr, cosr);

    %teta(pitch)
    sinp = 2 * (q_0 * q_2 - q_3 * q_1);
    if (abs(sinp) >= 1)
        teta = sign(sinp)*(pi/2); % use 90 degrees if out of range
    else
        teta = asin(sinp);
    end

    %phi(yaw)
    siny = 2.0 * (q_0 * q_3 + q_1 * q_2);
    cosy = 1.0 - 2.0 * (q_2 * q_2 + q_3 * q_3);
    psi = atan2(siny, cosy);
end
```

- **Angles d'Euler à quaternions**

Dans la convention de rotation z-y-x la multiplication des quaternions représentant chaque rotations intermédiaire résultent en la formule implémentée :

```
function[q] = from_Eul_to_quat_v(phi,teta,psi)
    cphi = cos(phi/2);
    sphi = sin(phi/2);
    cteta = cos(teta/2);
    steta = sin(teta/2);
    cpsi = cos(psi/2);
    spsi = sin(psi/2);

    q_0 = cphi * cteta * cpsi + sphi * steta * spsi;
    q_1 = cphi * steta * cpsi - sphi * cteta * spsi;
    q_2 = cphi * cteta * spsi + sphi * steta * cpsi;
    q_3 = sphi * cteta * cpsi - cphi * steta * spsi;

    q=[q_0;-q_1;-q_2;-q_3];
end
```

- **Angles d'Euler à matrice de rotation**

En implémentant l'équation (eq.3) on obtient :

```
function[R] = from_eul_to_rotMat(phi,teta,psi)
    %Calculate rotation about x axis
    R_x(:, :, k) = [1 0 0;
                   0 cos(phi) sin(phi);
                   0 -sin(phi) cos(phi)];

    %Calculate rotation about y axis
    R_y(:, :, k) = [cos(teta) 0 -sin(teta);
                   0 1 0;
                   sin(teta) 0 cos(teta)];

    %Calculate rotation about z axis
    R_z(:, :, k) = [cos(psi) sin(psi) 0;
                   -sin(psi) cos(psi) 0;
                   0 0 1];

    %Combined rotation matrix
    R(:, :, k) = R_x(:, :, k)*R_y(:, :, k)*R_z(:, :, k);
end
```

- **Quaternions à matrice de rotation**

On a

$$R = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2q_1q_2 - 2q_0q_3 & 2q_1q_3 + 2q_0q_2 \\ 2q_1q_2 + 2q_0q_3 & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2q_2q_3 - 2q_0q_1 \\ 2q_1q_3 - 2q_0q_2 & 2q_2q_3 + 2q_0q_1 & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix}$$

```
function[R] = from_quat_to_rotMat(q)

    q_0 = q(1); %a
    q_1 = q(2); %b
    q_2 = q(3); %c
    q_3 = q(4); %d
```



```

    qv = [q_1;q_2;q_3];

    R = [q_0*q_0+q_1*q_1-q_2*q_2-q_3*q_3    2*q_1*q_2-2*q_0*q_3
2*q_1*q_3+2*q_0*q_2;
        2*q_1*q_2+2*q_0*q_3    q_0*q_0-q_1*q_1+q_2*q_2-q_3*q_3
2*q_2*q_3-2*q_0*q_1;
        2*q_1*q_3-2*q_0*q_2    2*q_2*q_3+2*q_0*q_1    q_0*q_0-q_1*q_1-
q_2*q_2+q_3*q_3];
end

```

- **Matrice de rotation à quaternion**

```

function[q] = from_rotMat_to_quat(R)

    %R = r11 r12 r13
    %   r21 r22 r23
    %   r31 r32 r33

    r11 = R(1,1); %m00
    r12 = R(1,2); %m01
    r13 = R(1,3); %m02
    r21 = R(2,1); %m10
    r22 = R(2,2); %m11
    r23 = R(2,3); %m12
    r31 = R(3,1); %m20
    r32 = R(3,2); %m21
    r33 = R(3,3); %m22

    if (r33 < 0)
        if (r11 > r22)
            t= 1 + r11 -r22 -r33;
            q = [r23-r32;-t;-(r12+r21);-(r31+r13)];
        else
            t= 1 -r11 + r22 -r33;
            q = [r31-r13; -(r12+r21); -t; -(r23+r32)];
        end
    else
        if (r11 < -r22)
            t= 1 -r11 -r22 + r33;
            q = [r12-r21; -(r31+r13); -(r23+r32); -t];
        else
            t= 1 + r11 + r22 + r33;
            q = [t;-(r23-r32);-(r31-r13);-(r12-r21)];
        end
    end
    q= q*0.5 / sqrt(t);
end

```

## 2- Algorithmes d'estimation à partir des mesures uniquement

L'approche la plus simple à laquelle on peut penser pour estimer la pose (pose = position et orientation) du solide (détecteur ou source) est de se baser sur les mesures des capteurs uniquement. La première approche dans ce sens est d'utiliser les mesures du gyromètre pour estimer l'orientation et la mesure de l'accéléromètre pour estimer la position dans le cadre de la **navigation est l'estime**. La deuxième approche dans ce sens est de combiner les données des 2 capteurs pour l'estimation de la pose dans le cadre d'un **filtre complémentaire**.

### a- Navigation à l'estime

Elle consiste à déduire la position d'un mobile à partir de sa dernière position connue en utilisant la relation existante entre l'accélération relative du solide mesurée par l'accéléromètre  ${}^b\mathbf{a}_m$  et sa position estimée  ${}^w\mathbf{p}$ . L'orientation  ${}^bq$  est estimée à partir de la dernière orientation et de la vitesse angulaire mesurée par les gyromètres  ${}^b\omega$ . Le principe est illustré sur la Fig.4 :

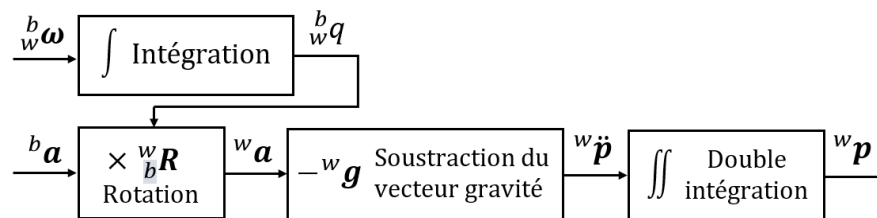


Fig. 4 : Schéma bloc pour la navigation à l'estime.

Dans les cas idéal (mesures des capteurs dénuées d'erreurs), la navigation à l'estime donnerait des estimés de la position et de l'orientation parfaits. Malheureusement, les mesures sont entachées d'erreurs *Seq.III-1*. Plus les mesures sont bonnes, plus la navigation à l'estime donne de bons résultats.

### Exemples sur données simulées :

**J'utilise de l'ordre 1 ici. Il faut que j'implémente de l'ordre 2...**

## b- Filtre complémentaire

Nous pouvons également estimer le quaternion de l'orientation  ${}^b_w q$  à partir des données provenant des accéléromètres triaxiaux et des gyromètres triaxiaux.

Le principe est de considérer que les mesures basse fréquence de l'accéléromètre (sujet à du bruit haute fréquence) et que les mesures haute fréquence des gyromètres (sujet à une dérive basse fréquence) sont précises. Par conséquent, au lieu de traiter indépendamment les mesures du gyromètre et l'accéléromètre, nous pouvons les combiner pour l'estimation de l'orientation. Pour ce faire, le principe est de filtrer par un passe-haut la sortie du gyromètre et de filtrer par un passe bas les mesures de l'accéléromètre. Ensuite, nous interpolons les données filtrées pour estimer le quaternion de l'orientation. En pratique, le filtre complémentaire est implémenté selon la méthode proposée dans [ref https://www.mdpi.com/1424-8220/15/8/19302/pdf](https://www.mdpi.com/1424-8220/15/8/19302/pdf).

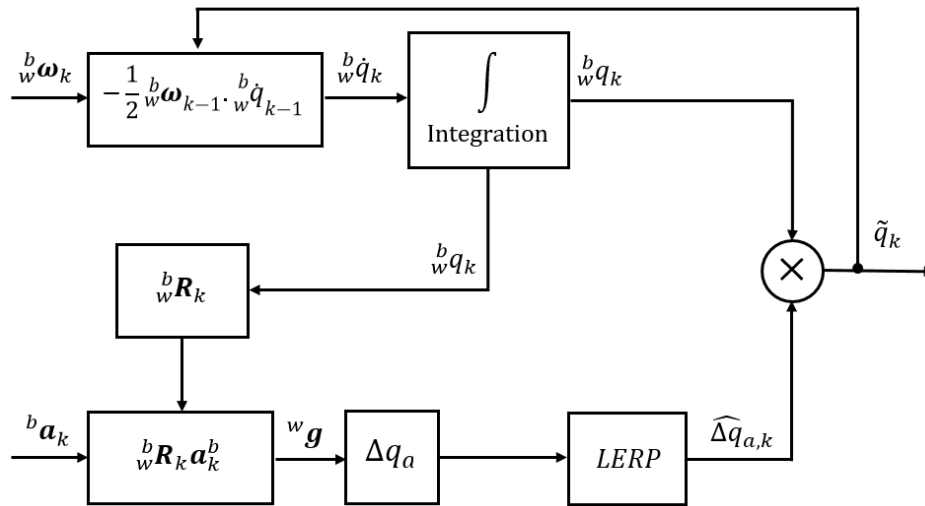


Fig. 5 : Schéma bloc pour le filtre complémentaire

Nos fréquences d'échantillonnage sont suffisamment grandes pour utiliser une simple interpolation linéaire (LERP) :

$$\widehat{\Delta q}_{acc} = (1 - \alpha)q_I + \alpha \Delta q_{acc}$$

Ici  $\alpha \in [0,1]$  est notre paramètre de réglage du filtre complémentaire. Il permet de donner plus ou moins du poids aux mesures de l'accéléromètre. Au vu de nos temps d'intégration courts les meilleurs résultats sont obtenus pour une valeur de  $\alpha \approx 1$ . L'intérêt du filtre complémentaire est très limité pour les courts temps d'intégration.

### Exemple sur données simulées :



### 3- Algorithmes d'estimation à partir des mesures et de modèles

Afin d'être plus robustes vis-à-vis des erreurs de mesures, il paraît judicieux de prendre en compte un modèle dynamique dans l'estimation de pose. En effet, nous disposons d'une bonne connaissance à priori du C-arm (trajectoire proche d'une trajectoire circulaire, vitesses angulaires faibles, temps d'intégration courts...). Nous allons prendre en compte ce modèle dans le cadre d'un **filtre de Kalman étendu (EKF)**.

#### a- Choix du filtre de Kalman étendu additif contraint

La première question à se poser est : pourquoi le choix d'un filtre de Kalman étendu ? L'idée découle directement de l'approche la plus directe pour la résolution d'un problème non linéaire : linéariser autour de la meilleure estimation actuelle puis appliquer un algorithme optimal d'estimation pour un système linéaire. C'est le filtre de Kalman étendu.

**Plusieurs implémentations sont possibles pour le filtre de Kalman étendu :**

- EKF minimal: Filtre de Kalman étendu qui prend comme entrée des angles d'Euler. Cependant, il est affecté par les singularités de ces derniers. Convient si les rotations à estimer sont petites et ne s'éloignent pas d'une orientation de référence. Ceci n'est pas le cas pour le C-arm.
- EKF multiplicative (MEKF) 6 et 7 : l'orientation est un produit de l'orientation prédite et d'une rotation issue de la déviation par rapport à l'estimé **ref 5-7 82-84**

$$q = \delta q(\Phi) * \hat{q}$$

$\hat{q}$  : quaternion estimé

$\delta q$  : rotation de l'estimé à la vraie valeur  $q$  par 3 angles  $\Phi = (\varphi, \theta, \psi)$

Les avantages de cette représentation est que

- $\hat{q}$  est toujours un quaternion unitaire
  - La matrice de covariance est de dimension minimum
  - L'angle estimé ne présente pas de singularité
- EKF additif : Peut être contraint ou non et s'il est contraint correctement équivalent à MEKF. S'il n'est pas contraint, il assouplit la normalisation et les matrices de covariance calculées incluent alors l'incertitude de non-normalisation.

Des alternatives au filtre de Kalman additif contraint et des justifications de choix sont présentées en **Annexe 2**.

J'utilise un EKF additif contraint pour l'estimation de l'orientation : C'est plus simple à implémenter qu'un filtre de Kalman multiplicatif tout en profitant de ses avantages. C'est la méthode de choix pour une grande majorité des applications. Il s'agit d'un outil flexible et simple, avec un état de l'art fourni et pouvant intégrer une grande variété de mesures.

Limites : Peut ne pas converger si les modèles dynamique ou si les modèles de mesures sont hautement non linéaires ou en l'absence d'estimation a priori de l'état. (à priori pas le cas du C-arm)

## b- Principe du Filtre de Kalman étendu

**Principe :** Le filtre de Kalman (KF) fonctionne en deux phases distinctes : une **phase de prédiction** et une **phase de correction**. Cet algorithme fournit à chaque nouvelle observation une nouvelle estimation de la variable d'état. Le filtrage de Kalman utilise les observations dans l'ordre où elles sont disponibles, d'où l'appellation séquentielle. Par conséquent, l'estimation de l'état du système ne dépendra pas des observations futures mais uniquement des observations passées et présentes.

- Dans la **phase de prédiction**, la pose actuelle du solide est prédite à partir des anciennes poses selon des modèles dynamiques. Après cette étape, une nouvelle pose et une covariance de prédiction sont calculées. Dans un premier temps, nous supposons que notre **système est markovien** (l'information utile pour la prédiction du futur est entièrement contenue dans l'état présent du processus et n'est pas dépendante des états antérieurs : le système n'a pas de « mémoire »). Par conséquent, la pose actuelle est prédite à partir de la pose à l'instant précédent uniquement.
- Ensuite, au cours de la **phase de correction**, les mesures des capteurs inertiels et leurs incertitudes sont utilisées pour corriger la prédiction. Ces incertitudes et la covariance de prédiction déterminent l'incidence de la mesure sur la prédiction. Le filtrage est décrit dans le diagramme Fig. 6.

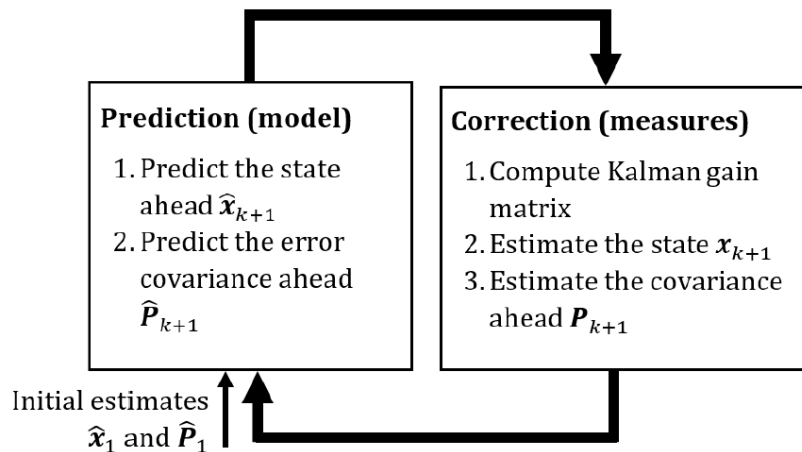


Fig. 6 : Résumé des deux étapes d'un filtrage de Kalman

## c- Deux KF différents pour l'orientation et la position

Nos temps d'acquisition sont courts ( $\sim 1$ min), nous notons expérimentalement que les biais du gyromètre ne dérivent pas pendant (écart types comparables pour des acquisitions statiques du gyromètre de 10s, 30s, 1min de l'ordre de  $0.0009$  °/s pour l'ensemble des axes). Cette durée est trop courte pour observer une dérive angulaire du gyromètre. De plus, une simple application permet de montrer que  $|\mathbf{a}_{coriolis}^w| = |-2\boldsymbol{\omega}_{n/terre} \wedge \mathbf{v}_M^R| \sim 10^{-4} m/s^2$  avec  $\boldsymbol{\omega}_{w/terre} = 7,292\ 115 \times 10^{-5}$  rad/s vitesse angulaire de la rotation de la terre et  $\mathbf{v}_M^R = 0.25$  m/s une valeur type de vitesse de notre mobile. On observe un écart type de bruit de mesure de l'ordre de  $0.03$   $m/s^2$ . L'accélération de Coriolis est comprise dans nos bruits de mesures est nous pouvons poser  $\mathbf{a}_{coriolis}^n = 0$ . Il apparait, donc inutile de coupler les mesures des accéléromètres et des

gyromètres dans le cas du mouvement du C-arm. Nous allons implémenter 2 filtres de Kalman différents pour la position et pour l'orientation.

#### d- Implémentation du filtre de Kalman

Afin de simplifier les notations de cette sous partie, nous omettons les indices et les exposants, par exemple  ${}^w\mathbf{a}_k$  s'écrit  $\mathbf{a}_k$  et  ${}^d\boldsymbol{\omega}_k$  s'écrit  $\boldsymbol{\omega}_k$ .

On suppose qu'on a le système non linéaire suivant :

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \boldsymbol{\theta}_{k-1}, \mathbf{u}_k) + \mathbf{e}_{k-1}$$

$$\mathbf{y}_k = h(\mathbf{x}_k, \boldsymbol{\theta}_{k-1}) + \mathbf{n}_k$$

Ici, pour un échantillon  $k$ ,  $\mathbf{x}_k$  est le vecteur d'état composés des paramètres  $\boldsymbol{\theta}_k$  qu'on veut estimer,  $\mathbf{u}_k$  est le terme de commande,  $\mathbf{y}_k$  est le vecteur de mesure.

$f$  est la fonction d'état (qui décrit le système) et  $h$  la fonction de mesure (qui met en relation l'état et la mesure). Elles peuvent être linéaires ou non.

Les bruits d'état et de mesure sont représentés par  $\mathbf{e}_k \sim \mathcal{N}(0, \mathbf{Q}_k)$  et  $\mathbf{n}_k \sim \mathcal{N}(0, \mathbf{R}_k)$  respectivement où  $\mathbf{Q}_k$  et  $\mathbf{R}_k$  sont les matrices de covariance du bruit d'état et de mesure respectivement. Ils sont supposés indépendants.

Pour rappel, un filtre de Kalman a pour utilité d'estimer **la valeur d'un vecteur d'état  $\mathbf{x}$**  ainsi que la précision avec laquelle ce vecteur d'état a été prédit en estimant **une matrice de covariance de l'estimation**. Les équations des étapes de prédiction et de correction se présentent ainsi :

#### Prédiction :

- 1- On prédit le vecteur d'état à partir de l'estimé précédent  $\hat{\mathbf{x}}_{k-1|k-1}$ , de **modèle dynamiques** et du terme de **commande  $\mathbf{u}_k$** :

$$\hat{\mathbf{x}}_{k|k-1} = f(\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_k)$$

- 2- On calcule la matrice de covariance de la prédiction :

$$\mathbf{P}_{k|k-1} = \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^t + \mathbf{Q}_k$$

avec  $\mathbf{Q}_k$  paramètre d'entrée et de réglage du Kalman, elle sera calculée à partir des bruits de mesure des IMUs et des écarts types d'erreur de calibration, matrice de covariance du modèle et  $\mathbf{F}_k = \left. \frac{\partial f}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_k}$

#### Correction :

- 3- On calcule l'innovation  $\tilde{\boldsymbol{\chi}}_k$

$$\tilde{\boldsymbol{\chi}}_k = \mathbf{z}_k - h(\hat{\mathbf{x}}_{k|k-1})$$

où  $\mathbf{z}_k$  le vecteur de mesures à l'instant  $k$

$h$  fonction de mesure

- 4- On calcule la covariance de l'Innovation  $\mathbf{S}_k$ :

$$\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^t + \mathbf{R}_k$$

où :  $F_k = \frac{\partial h}{\partial x} \Big|_{\hat{x}_{k|k-1}}$  et  $R_k$  matrice de covariance de bruit de mesures issue des écarts types de bruit de mesure de la caméra

5- On calcule le gain de Kalman

$$K_k = P_{k|k-1} H_k^t S_k^{-1}$$

6- On met à jour le vecteur d'état

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \tilde{y}_k$$

7- On calcule la matrice de covariance de l'estimé

$$P_{k|k} = (I - K_k H_k) P_{k|k-1}$$

Il ne nous reste plus qu'à déterminer les différents termes dans notre problème pour l'orientation et pour la position.

	Orientation	Position
vecteur de commande	$u_a = 0$	$u_\omega = 0$
vecteur d'état	$x_a = [p \ v \ a]^t$ Composé du vecteur de position $p$ , du vecteur vitesse $v$ et du vecteur accélération $a$	$x_\omega = [q \ \omega]^t$ Composé du quaternion de l'orientation $q_k$ et de la vitesse angulaire $\omega$
fonction d'état en régime continu	$\frac{\partial v(t)}{\partial t} = a(t)$ $\frac{\partial p(t)}{\partial t} = v(t)$ $a(t) = -\omega^2(t)p(t)$ <p>Les deux premières équations sont des relations classiques de la dynamique. La troisième équation renvoie à l'expression de l'accélération pour un mouvement circulaire uniforme. C'est le modèle que nous choisissons au vu de notre connaissance à priori sur le C-arm. Il donne de meilleurs résultats expérimentaux sur table tournante que les modèle accélération uniforme et accélération nulle.</p>	$\frac{\partial q(t)}{\partial t} = -\frac{1}{2}\Omega(t)q(t)$ $\frac{\partial \omega(t)}{\partial t} = 0$ <p>Pour plus de détails sur la première équation, voir la partie quaternions. La deuxième équation renvoie au choix de modèle : vitesse angulaire constante</p>



fonction d'état en régime discret	Nous choisissons d'échantillonner au second ordre, pour plus d'informations sur ce choix voir la partie suivante.	
	$\mathbf{v}_{k+1} = \Delta f\left(t_k + \frac{\Delta t}{2}, \mathbf{v}_k + \frac{\Delta t}{2} \mathbf{a}_k\right)$ $\mathbf{p}_{k+1} = \Delta f\left(t_k + \frac{\Delta t}{2}, \mathbf{p}_k + \frac{\Delta t}{2} \mathbf{v}_k\right)$ <p>Ici  <math>\frac{\partial \mathbf{p}(t)}{\partial t} = f(t, \mathbf{p}(t))</math> et <math>t_k = t_0 + k\Delta t</math> avec <math>t_0</math>  l'instant initial et <math>\frac{\partial \mathbf{v}(t)}{\partial t} = f(t, \mathbf{v}(t))</math>  Et</p> $\mathbf{a}_{k+1} = -\boldsymbol{\omega}_k^2 \mathbf{p}_k$	$q_{k+1} = \exp(\boldsymbol{\Omega}_k \Delta t) q_k$ Après échantillonnage au second ordre : $q_{k+1} = \left( \mathbf{I} + \frac{1}{2} \boldsymbol{\Omega}_k \Delta t - \frac{1}{2!} \left(\frac{1}{2} \boldsymbol{\Omega}_k \Delta t\right)^2 \right) q_k$ Et : $\boldsymbol{\omega}_{k+1} = \boldsymbol{\omega}_k$
Matrice de bruit du processus	$\mathbf{Q}_{a,k}$ et $\mathbf{Q}_{\omega,k}$ sont des matrices de réglage des EKF d'estimation de position et d'orientation respectivement. Afin de la dimensionner, on évalue empiriquement l'erreur commise lors de la construction du modèle (par exemple pour la table tournante on construit $\mathbf{Q}_{a,k}$ et $\mathbf{Q}_{\omega,k}$ comme les matrices diagonales dont les termes des diagonales sont la somme des écarts types de la différence entre la trajectoire/l'orientation de la table tournante et une trajectoire/une orientation circulaire idéale et des erreurs dus à l'échantillonnage <i>Seq.IX-2.</i> )	
Vecteur et équation de mesure	On reprend l'équation de mesure de l'accéléromètre : $\mathbf{y}_{a,k} = \mathbf{M}'_a \mathbf{R}_{n,k}^b ({}^n \mathbf{a}_k - \mathbf{g}) + {}^b \mathbf{b}'_a + {}^b \mathbf{n}_{a,k}$ <p>Ici le vecteur de mesure est <math>\mathbf{y}_{a,k}</math></p>	On reprend l'équation de mesure du gyromètre $\mathbf{y}_{\omega,k} = \mathbf{M}'_\omega {}^b \boldsymbol{\omega}_k + {}^b \mathbf{b}'_\omega + {}^b \mathbf{n}_{\omega,k}$ <p>Ici le vecteur de mesure est <math>\mathbf{y}_{\omega,k}</math></p>
Matrice de bruit de mesure	Directement issue des écart-types des bruits des capteurs <i>Seq.IX-2</i>	

### e- Initialisation

L'initialisation du filtre de Kalman consiste en la définition des vecteurs d'état  $\hat{\mathbf{x}}_{a,0}$  et  $\hat{\mathbf{x}}_{\omega,0}$ , et de la matrice  $\mathbf{P}_0$ . Nous choisissons

$$\hat{\mathbf{x}}_{a,0} = [\hat{\mathbf{p}}_0 \quad \hat{\mathbf{v}}_0 \quad \hat{\mathbf{a}}_0]^t = [\hat{\mathbf{p}}_0 \quad [0 \ 0 \ 0] \quad [0 \ 0 \ 0]]^t$$

avec  $\hat{\mathbf{p}}_0$  la position du module étudié tirée des côtes du dessin **en annexe 3** et :

$$\hat{\mathbf{x}}_{\omega,0} = [\hat{q}_0 \quad \hat{\boldsymbol{\omega}}_0]^t = [[1 \ 0 \ 0 \ 0] \quad [0 \ 0 \ 0]]$$

Nous choisissons  $\mathbf{P}_0 = \mathbf{Q}_0$

Nous discuterons d'une piste pour initialiser les capteurs (entre autres) en utilisant une caméra dans la *Seq.XI*.

## f- Echantillonnage

Dans cette sous partie nous allons discuter le problème d'échantillonnage d'une fonction continue d'abord, puis justifier le choix du second ordre pour l'échantillonnage de notre fonction.

On s'intéresse à la première dérivée :

$$\frac{\partial \mathbf{y}(t)}{\partial t} = f(t, \mathbf{y})$$

avec  $t \in [t_0 \ t_{final}]$  et  $y(t_0) = y_0$ ,  $f(t, \mathbf{y})$  est la dérivée de  $y(t)$

L'idée derrière cette partie est d'obtenir une approximation discrète de la dérivée : nous allons approximer la dérivée par une différence. Pour se faire, nous supposons que la période d'échantillonnage est suffisamment petite.

### 1- Méthode d'Euler

Notons, que nous allons uniquement développer la méthode d'obtention de la formule de la méthode d'Euler par formule de Taylor **ref** mais d'autres méthodes d'obtention (par définition de la dérivée et par intégration numérique) sont possibles.

Nous pouvons écrire :

$$y(t_i + \Delta t) = y(t_i) + \frac{y^{(1)}(t_i)}{1!} \Delta t + \frac{y^{(2)}(t_i)}{2!} \Delta t^2 + \dots + \frac{y^{(n)}(t_i)}{n!} \Delta t^n + o(\Delta t^n)$$

Ici,  $y^{(n)}$  la  $n$ -ième dérivée de  $y$  et  $o(\Delta t^n)$  un terme négligeable par rapport à  $\Delta t^n$  au voisinage de 0.

Pour obtenir la formule d'Euler à partir de la série de Taylor nous nous limitons à l'ordre 1 de cette dernière :

$$y(t_i + \Delta t) = y(t_i) + \frac{y^{(1)}(t_i)}{1!} \Delta t + o(\Delta t)$$

Ce qui donne

$$y(t_i + \Delta t) \approx y(t_i) + \Delta t y'(t_i)$$

Ce qui donne la formule de la méthode d'Euler :

$$y_{k+1} \approx y_k + \Delta t f(t_k, y_k)$$

### 2- Formules d'ordre supérieur (RungeKutta)

Il existe différents moyens d'extraire des formules de discrétisation d'ordre supérieur. Une idée intuitive serait d'utiliser un ordre  $n$  supérieur de la série de Taylor mais il faut que les fonctions soient dérivables à cet ordre ce qui n'est pas immédiat.

La deuxième idée est d'utiliser des méthodes linéaires à plusieurs étapes (Linear multistep methods) pour approximer  $\int_{t_i}^{t_{i+1}} f(t, y) dt$  ce qui engendre les méthodes de RungeKutta [ref https://fr.wikipedia.org/wiki/M%C3%A9thodes\\_de\\_Runge-Kutta](https://fr.wikipedia.org/wiki/M%C3%A9thodes_de_Runge-Kutta).

On écrit :

$$y(t_{k+1}) = y(t_k) + \int_{t_k}^{t_{k+1}} f(t, y(t)) dt = y(t_k) + \Delta t \int_0^1 f(t_k + u\Delta t, y(t_k + u\Delta t)) du$$

On calculera ces intégrales par une méthode de quadrature ici la méthode de quadrature de Gauss [ref https://fr.wikipedia.org/wiki/M%C3%A9thodes\\_de\\_quadrature\\_de\\_Gauss](https://fr.wikipedia.org/wiki/M%C3%A9thodes_de_quadrature_de_Gauss).

Nous allons expliciter les méthodes de Runge kutta d'ordre 2, 3 et 4

- **Ordre 2 : Méthode Midpoint (RK2)**

$$\begin{aligned} k_1 &= \Delta t f(t_k, y_k) \\ k_2 &= \Delta t f\left(t_k + \frac{\Delta t}{2}, y_k + \frac{1}{2}k_1\right) \\ y_{k+1} &= y_k + k_2 \end{aligned}$$

- **Ordre 3 : Méthode de Heun (RK3)**

$$\begin{aligned} k_1 &= \Delta t f(t_k, y_k) \\ k_2 &= \Delta t f\left(t_k + \frac{2}{3}\Delta t, y_k + \frac{2}{3}k_1\right) \\ y_{k+1} &= y_k + \frac{1}{4}k_1 + \frac{3}{4}k_2 \end{aligned}$$

- **Ordre 4 : Méthode Runge-kutta (RK4)**

$$\begin{aligned} k_1 &= \Delta t f(t_k, y_k) \\ k_2 &= \Delta t f\left(t_k + \frac{1}{2}\Delta t, y_k + \frac{1}{2}k_1\right) \\ k_3 &= \Delta t f\left(t_k + \frac{1}{2}\Delta t, y_k + \frac{1}{2}k_2\right) \\ y_{k+1} &= y_k + b_1k_1 + b_2k_2 + b_3k_3 \end{aligned}$$

### 3- Erreur de discrétisation

En échantillonnant une fonction, nous pouvons discerner deux types d'erreurs : les erreurs d'arrondi qui résultent du fait que notre machine de calcul a une précision finie, nous supposons que ces erreurs sont négligeables et les erreurs d'échantillonnage qui elles aussi peuvent être divisées en 2 catégories.

- **L'erreur globale d'échantillonnage** : L'erreur globale d'échantillonnage à l'instant  $t_i$  est la différence entre la valeur réelle  $y(t_k)$  et son approximation  $y_k$  :

$$|y(t_k) - y_k|$$

- **L'erreur locale d'échantillonnage** : L'erreur locale d'échantillonnage à l'instant  $t_i$  est l'erreur faite en supposant que le point de départ de l'estimation de  $y_{k+1}$  est la valeur exacte  $y(t_k)$  (et non son estimation  $y_k$ ). C'est l'erreur due uniquement à l'approximation d'une différentielle par une différence.

Si on illustre les deux erreurs sur la méthode d'Euler nous obtenons :

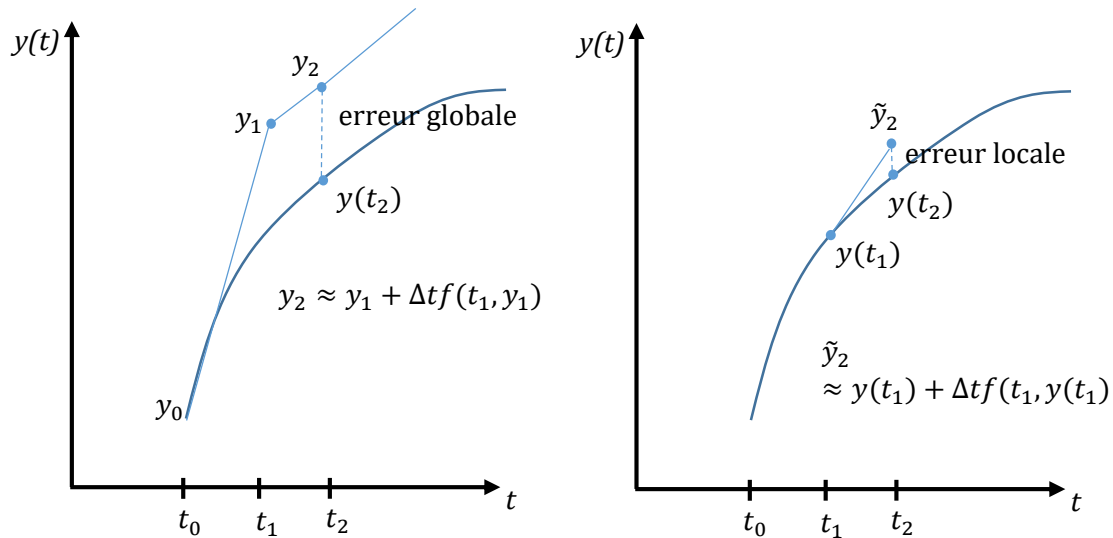


Fig. 7 : Illustration de l'erreur locale et de l'erreur globale sur la méthode d'Euler

Dans la suite nous allons développer l'expression des erreurs de discrétisation pour la méthode d'Euler pour estimer leur ordre puis présenter les ordres de grandeur des erreurs pour les autres ordres de discrétisation.

#### 4- Ordre de grandeur de l'erreur de discrétisation pour la méthode d'Euler

Afin de calculer l'erreur de discrétisation introduite par la méthode d'Euler, nous allons commencer par trouver l'expression de l'**erreur locale** d'échantillonnage en fonction de  $\Delta t$ . Ensuite nous utiliserons ce résultat pour calculer l'erreur globale.

Comme écrit plus tôt, l'erreur locale de troncature est la différence entre la solution exacte et l'erreur locale.

$$\tau_{k+1} = |y(t_{k+1}) - \tilde{y}_{k+1}|$$

avec

$$\tilde{y}_{k+1} = y(t_k) + \Delta t f(t_k, y(t_k))$$

en remplaçant

$$\tau_{k+1} = |y(t_{k+1}) - y(t_k) - \Delta t f(t_k, y(t_k))|$$

On peut simplifier cette expression en utilisant les séries de Taylor :

$$y(t_{k+1}) = y(t_k) + \Delta t y'(t_k) + \frac{\Delta t^2}{2} y''(\delta_k)$$

avec  $\delta_k \in [t_k, t_{k+1}]$

On a alors :

$$y(t_{k+1}) - \tilde{y}_{k+1} = \left[ y(t_k) + \Delta t y'(t_k) + \frac{\Delta t^2}{2} y''(\delta_k) \right] - y(t_k) - \Delta t f(t_k, y(t_k))$$

$$y(t_{k+1}) - \tilde{y}_{k+1} = \left[ y(t_k) + \Delta t f(t_k, y(t_k)) + \frac{\Delta t^2}{2} y''(\delta_k) \right] - y(t_k) - \Delta t f(t_k, y(t_k))$$

donc :

$$y(t_{k+1}) - \tilde{y}_{k+1} = \frac{\Delta t^2}{2} y''(\delta_k)$$

d'où :

$$\tau_{k+1} = \frac{\Delta t^2}{2} |y''(\delta_k)|$$

Nous avons  $y''(\delta_k)$  borné, donc

$$\tau_{k+1} = \mathcal{O}(\Delta t^2)$$

Sans démonstration formelle, nous pouvons comprendre que le cumul d'erreurs locales de l'ordre de  $\Delta t^2$  entraîne une erreur globale de l'erreur de  $\Delta t$ . En effet, si nous avons N

échantillons :  $\Delta t = \frac{t_{final} - t_0}{N}$ . Posons l'erreur locale  $\tau_{k+1} = C(\Delta t^2)$

$$Erreur\ globale(Euler) = N C(\Delta t^2) = \frac{t_{final} - t_0}{\Delta t} C(\Delta t^2) = \mathcal{O}(\Delta t)$$

1. L'erreur d'échantillonnage globale pour la méthode Midpoint (RK2) est de l'ordre de  $(\Delta t^2)$
2. L'erreur d'échantillonnage globale pour la méthode de Heun (RK3) est de l'ordre de  $(\Delta t^2)$
3. L'erreur d'échantillonnage globale pour la méthode Runge Kutta (RK4) est de l'ordre de  $(\Delta t^3)$

### 5- Choix de l'ordre d'échantillonnage

Supposons qu'on travaille avec une fréquence d'échantillonnage de 100Hz on a  $\Delta t = 0.01s$ ,  $\Delta t^2 = 10^{-4}s$  et  $\Delta t^3 = 10^{-6}s$

Nous allons faire une application numérique pour estimer l'erreur due à la discrétisation engendrée par la méthode d'Euler, la méthode Midpoint et la méthode RK4.

Nous allons supposer que nous générons un mouvement circulaire de vitesse  $10^\circ/s$ , ce qui entraîne une vitesse de 0.2m/s en considérant un rayon de trajectoire de 1.2m

Les erreurs engendrées par les différents ordres de discrétisations sont présentés dans le tableau :

	Euler	Midpoint	RK4
Distance	$\mathcal{O}(1mm)$	$\mathcal{O}(0.01mm)$	$\mathcal{O}(0.0001mm)$
orientation	$\mathcal{O}(0.1^\circ)$	$\mathcal{O}(0.001^\circ)$	$\mathcal{O}(10^{-5}^\circ)$

Comme nous pouvons le voir, pour une fréquence d'échantillonnage de 100Hz, les erreurs de discrétisation engendrées par la méthode d'Euler sont de même ordre de grandeur que la précision désirée, il paraît judicieux d'échantillonner notre modèle à un ordre supérieur. L'ordre 2 (méthode Midpoint) paraît suffisant. Au vu de la très petite erreur d'échantillonnage engendrée par le RK4, nous allons supposer que le signal échantillonné en RK4 représente notre signal continu.

Nous allons vérifier ces résultats théoriques sur une acquisition *Fig.9*. Nous réalisons une acquisition sur notre table tournante d'une rotation continue de 120° à 500Hz. Nous choisissons un modèle dynamique de rotation à vitesse angulaire constante et nous échantillonons ce modèle grâce à la méthode d'Euler (a), la méthode midpoint (b) et la méthode RK4 (c) nous affichons le résultat obtenu pour les estimés de position par notre filtre de Kalman pour l'axe x sur la *Fig.8* et pour l'ensemble des axes sur le *Tab.4*. Nous supposons que les dimensions obtenues grâce à l'algorithme RK4 sont la référence. Nous donnons le même poids à la mesure et au modèle dans notre Kalman.

Ici refaire l'étude pour acquisition à 100Hz avec comparaison avec RK4 et simulé.

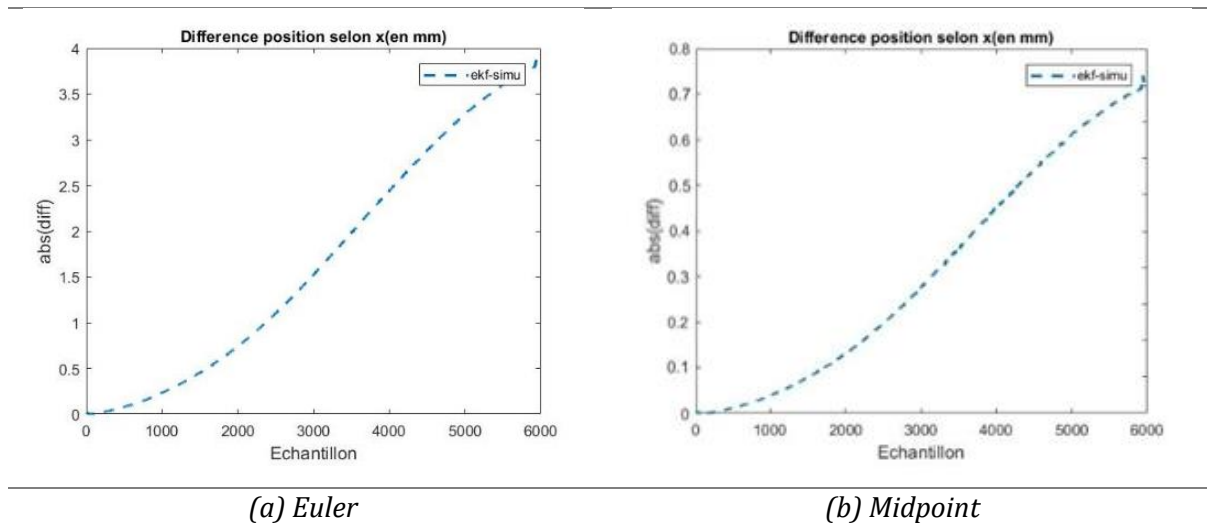
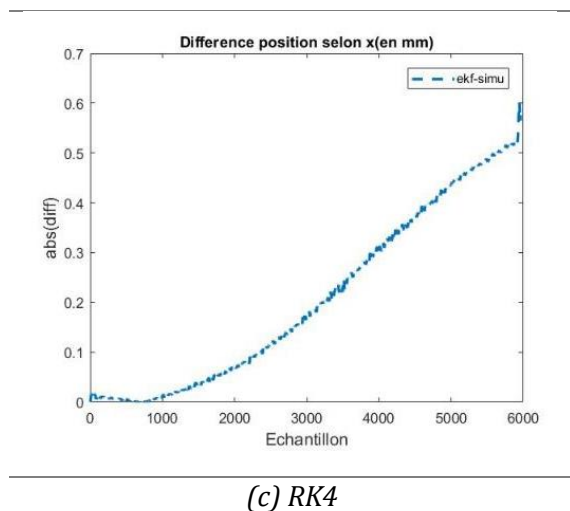


Fig. 8 : Erreurs des différents algorithmes de discrétisation



	X	Y
Euler	3,70mm	4,42mm
Midpoint	0,7mm	1,12mm
RK4	0,51mm	0,73mm

## V- Calcul des paramètres de la matrice de projection

Après l'estimation, nous disposons à chaque instant k de :

- a- la position du détecteur exprimée dans le repère monde  ${}^d\mathbf{p}$  et l'orientation du détecteur  ${}^w_dq / [{}^d\varphi \quad {}^d\theta \quad {}^d\psi]$  du repère lié au détecteur {d} dans le repère navigation {w}
- b- la position de la source exprimée dans le repère monde  ${}^s\mathbf{p}$  et l'orientation  ${}^w_sq / [{}^s\varphi \quad {}^s\theta \quad {}^s\psi]$  du repère lié à la source {s} dans le repère navigation {w}

Le but est de remonter aux différents paramètres de la matrice de projection que nous définissons comme :

- Les orientations  $[{}^d\varphi \quad {}^d\theta \quad {}^d\psi]$  sont directement récupérées en sortie des estimateurs, éventuellement après conversion de  ${}^w_dq_k$  en angles d'Euler
- Pour récupérer les translations :
  - Nous calculons les matrices de rotations  ${}^d_w\mathbf{R}$  et  ${}^s_w\mathbf{R}$  issues de  ${}^w_dq$  et à  ${}^w_sq$  respectivement.
  - Nous multiplions les vecteurs positions  ${}^d\mathbf{p}$  et  ${}^s\mathbf{p}$  par  ${}^d_w\mathbf{R}$  et  ${}^s_w\mathbf{R}$  respectivement, afin de les exprimer dans le repère correctement orienté à l'instant k :

$$\begin{bmatrix} dx_k \\ dy_k \\ swd_k - sdd_k \end{bmatrix} = {}^d_w\mathbf{R} {}^d\mathbf{p}$$

$$\begin{bmatrix} sx_k \\ sy_k \\ swd_k \end{bmatrix} = {}^s_w\mathbf{R} {}^s\mathbf{p}$$

- Nous calculons  $sdd_k$

**%Vérifier pour des «-« éventuels.**

```

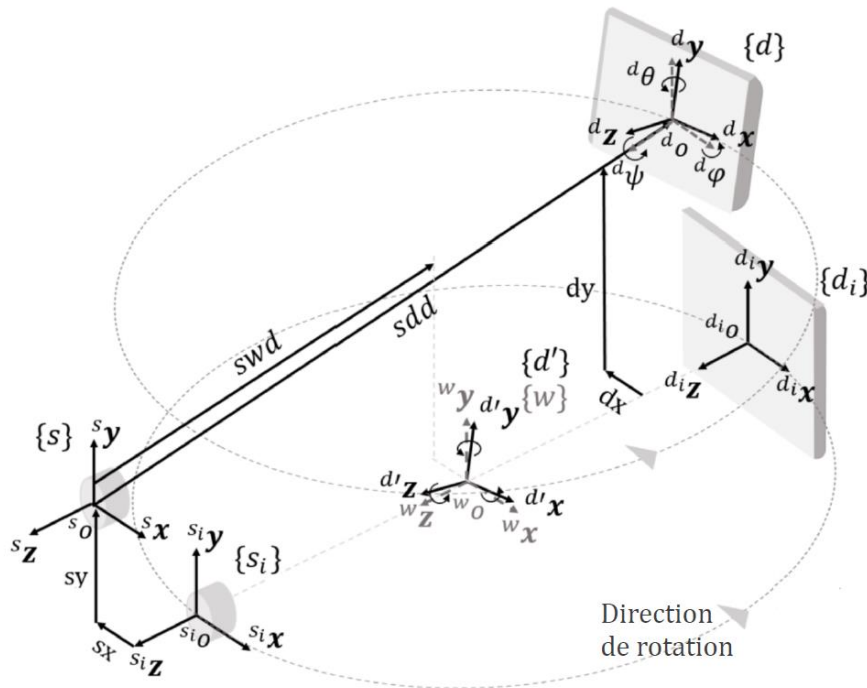
% Détecteur
r_dis_det = rot_carm_det(:, :, k) * pos_centre_det(:, k);
dx(k) = r_dis_det(1);
dy(k) = r_dis_det(2);
dwd(k) = - r_dis_det(3);           %dxd=sdd-swd

% Source
r_dis_src = rot_carm_src(:, :, k) * pos_centre_src(:, k); %
sx(k) = r_dis_src(1);
sy(k) = r_dis_src(2);
swd(k) = r_dis_src(3);
  
```

# VI- Calcul de la matrice de projection

## 1- Idée générale

Dans cette partie nous allons exploiter le repère détecteur virtuel  $\{d'\}$  présenté dans le schéma figure X. Pour plus de simplicité nous rappelons le schéma Fig. 1 :



Rappel Fig.1

La géométrie projective du C-arm est modélisée par un sténopé. L'espace de projection 3D est représenté par des vecteurs de 4 coordonnées homogènes  $x = [x_1 \ x_2 \ x_3 \ x_4]^t$  dont une au moins est non nulle. Les coordonnées homogènes sont très pratiques dans le cadre de la géométrie projective. En effet, elles permettent de représenter sous forme matricielle l'ensemble des transformations (rotation, translation, projection, symétrie...). Dans le cadre de ce modèle la géométrie projective du C-arm est représentée par une Matrice homogène de projection  $P$  de dimension  $(3 \times 4)$ .

Comme montré dans le Schéma figure X. Afin de calculer les coordonnées du projeté 2D  ${}^d p_{2D}$  d'un point 3D  ${}^w p_{3D}$  sur le plan détecteur 4 transformation sont nécessaires :

$$\begin{array}{ccccccc}
 {}^w p_{3D} & \xrightarrow{1} & {}^{d'} p_{3D} & \xrightarrow{2} & {}^s p_{3D} & \xrightarrow{3} & {}^s p_{2D} & \xrightarrow{4} & {}^d p_{2D} \\
 & \text{Rotation} & & \text{Translation} & & \text{Projection} & & \text{Translation} & \\
 & \underbrace{{}^d R_w}_{d'} & & \underbrace{{}^s s_0}_{d'} & & & & \underbrace{{}^d s_0}_{d'} & \\
 & & & & & & & & 
 \end{array}$$

note :

- ${}^y_x p_{3D}$ : Le point 3D solide du repère  $\{x\}$  exprimé dans le repère  $\{y\}$
- ${}^y_x p_{2D}$ : La projection 2D solide du repère  $\{x\}$  exprimé dans le repère  $\{y\}$



## 2- Détails

### a- Changement de repère de navigation {w} au repère détecteur virtuel {d'}

D'après la définition du repère {d'}, le changement de repère de {w} vers {d'} correspondant à une rotation d'angles  $[{}^d\varphi \quad {}^d\theta \quad {}^d\psi]^t$ . Nous pouvons écrire :

$${}^d_w \mathbf{p}_{3D} = {}^d_w \mathbf{R}_w \mathbf{p}_{3D}$$

En coordonnées homogènes cela donne :

$$\begin{bmatrix} {}^d_w x \\ {}^d_w y \\ {}^d_w z \\ 1 \end{bmatrix}_{3D} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} {}^w_w x \\ {}^w_w y \\ {}^w_w z \\ 1 \end{bmatrix}_{3D}$$

$$\text{On pose : } {}^d_w \mathbf{R}_{3 \times 3} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix} = [r'_1 \quad r'_2 \quad r'_3]$$

#### Note :

Du fait de l'ambiguïté de la représentation matricielle, les  $r_i$  (lignes de  $R_{3 \times 3}$ ) et  $r'_i$  (colonnes de  $R_{3 \times 3}$ ) et peuvent avoir une double signification selon que l'on considère un changement de repère ou une rotation dans un même repère :

- **Changement de repère** :  $r_i$  sont les coordonnées des vecteurs unitaires orthogonaux représentant l'orientation de la base {d'} dans la base {w}.
- **Changement de repère** :  $r'_i$  sont inversement les coordonnées des vecteurs unitaires orthogonaux représentant l'orientation de la base {w} dans la base {d'}
- **Rotation dans un même repère** :  $r_i$  sont les coordonnées des rotations des vecteurs unitaires orthogonaux représentant l'orientation de la base {d'} dans la base {d'}.
- **Rotation dans un même repère** :  $r'_i$  sont les coordonnées des rotations des vecteurs unitaires orthogonaux représentant l'orientation de la base {w} dans la base {w}.

### b- Changement de repère du détecteur virtuel {d'} vers le repère source {s}

Un changement de repère de {d'} vers {s} correspond à une translation de vecteur  $\overrightarrow{d'o'so} = \begin{bmatrix} sx \\ sy \\ sid \end{bmatrix}$ . Les coordonnées de  ${}^d_w \mathbf{p}_{3D}$  dans {d'} deviennent alors  ${}^s_w \mathbf{p}_{3D}$  exprimées dans le référentiel lié à la source {s} :

$${}^s_w \mathbf{p}_{3D} = {}^s_d \mathbf{T} \cdot {}^d_w \mathbf{p}_{3D}$$

avec :

$${}^s_d \mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & -sx \\ 0 & 1 & 0 & -sy \\ 0 & 0 & 1 & -sid \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

### c- Projection du point de {w} sur le plan détecteur {d}

Une projection perspective telle qu'un point  ${}^w\mathbf{p}_{3D}$  de coordonnées  $[{}^w x \quad {}^w y \quad {}^w z]^t_{3D}$  exprimé dans {s} aura pour projeté sur le plan détecteur  ${}^d\mathbf{p}_{2D}$ . Cette projection se fait selon la direction  ${}^s\mathbf{z}$ . On écrit :

$${}^d\mathbf{p}_{2D} = \mathbf{H} \cdot {}^s\mathbf{p}_{3D}$$

avec  $\mathbf{H}$  matrice de projection de rapport de projection  $\frac{f}{z_s}$  ; la distance de projection est la « focale c'est-à-dire la distance source détecteur « sdd ».

En coordonnée homogène, la matrice H s'exprime par

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{sdd} & 0 \end{bmatrix}$$

### d- Changement de repère du repère source {s} au repère détecteur {d}

Un changement de repère de {s} vers {d} correspond à une translation de vecteur  $\overrightarrow{ppO_D}$ . On écrit :

$${}^d\mathbf{p}_{2D} = {}^d\mathbf{T} {}^s\mathbf{p}_{2D}$$

En se référant à la géométrie du système on retrouve :

$${}^d\mathbf{T} = \begin{bmatrix} 1 & 0 & sx - dx \\ 0 & 1 & sy - dy \\ 0 & 0 & 1 \end{bmatrix}$$

## 3- Transformation globale

On obtient la transformation globale suivante :

$${}^d\mathbf{p}_{2D} = \mathbf{P} {}^w\mathbf{p}_{3D}$$

avec

$$\mathbf{P} = {}^d\mathbf{T} \mathbf{H} {}^s\mathbf{T} {}^w\mathbf{R}$$

$$\mathbf{P} = \begin{bmatrix} 1 & 0 & sx - dx \\ 0 & 1 & sy - dy \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{sdd} & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -sx \\ 0 & 1 & 0 & -sy \\ 0 & 0 & 1 & -swd \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

En décomposition extrinsèque/intrinsèque :

$$\mathbf{P} = \mathbf{I} \mathbf{E} = \begin{bmatrix} 1 & 0 & \frac{sx - dx}{sdd} & 0 \\ 0 & 1 & \frac{sy - dy}{sdd} & 0 \\ 0 & 0 & \frac{1}{sdd} & 0 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & -sx \\ r_{21} & r_{22} & r_{23} & -sy \\ r_{31} & r_{32} & r_{33} & -swd \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

I étant définie à un facteur près, on peut écrire :

$$P = I E = \begin{bmatrix} sdd & 0 & sx - dx & 0 \\ 0 & sdd & sy - dy & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & -sx \\ r_{21} & r_{22} & r_{23} & -sy \\ r_{31} & r_{32} & r_{33} & -swd \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$P =$

$$\begin{bmatrix} sdd r_{11} + (sx - dx)r_{31} & sdd r_{12} + (sx - dx)r_{32} & sdd r_{13} + (sx - dx)r_{33} & -sdd sx - swd(sx - dx) \\ sdd r_{21} + (sy - dy)r_{31} & sdd r_{22} + (sy - dy)r_{32} & sdd r_{23} + (sy - dy)r_{33} & -sdd sy - swd(sy - dy) \\ r_{31} & r_{32} & r_{33} & -swd \end{bmatrix}$$

## VII- Propagation d'erreur

### 1- Formule globale de propagation d'Erreur

Nous supposons que les erreurs sont non couplées. Nous avons :

$$\mathbf{P} = f(sdd, sid, sx, sy, dx, dy, {}^a\varphi, {}^a\theta, {}^a\psi)$$

et l'incertitude sur la matrice de projection  $\mathbf{P}$  a pour borne supérieure :

$$\delta\mathbf{P} \leq \left| \frac{\partial f}{\partial sdd} \right| \delta sdd + \left| \frac{\partial f}{\partial sid} \right| \delta swd + \left| \frac{\partial f}{\partial sx} \right| \delta sx + \left| \frac{\partial f}{\partial sy} \right| \delta sy + \left| \frac{\partial f}{\partial dx} \right| \delta dx + \left| \frac{\partial f}{\partial dy} \right| \delta dy \\ + \left| \frac{\partial f}{\partial sx} \right| \delta sx + \left| \frac{\partial f}{\partial {}^a\varphi} \right| \delta {}^a\varphi + \left| \frac{\partial f}{\partial {}^a\theta} \right| \delta {}^a\theta + \left| \frac{\partial f}{\partial {}^a\psi} \right| \delta {}^a\psi$$

Pour simplifier la notation. Comme seule l'orientation du détecteur intervient dans le calcul de la matrice de projection, nous simplifions les notations des angles en supprimant les exposants  ${}^a\varphi$ ,  ${}^a\theta$  et  ${}^a\psi$  sont notés  $\varphi$ ,  $\theta$  et  $\psi$  respectivement. Dans ce qui suit nous allons donner les formules des dérivées partielles :

### 2 - Dérivée de la matrice de projection par rapport aux distances

#### a- Dérivée par rapport à sdd

$$\frac{\partial f}{\partial sdd} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & -sx - \frac{\partial sid}{\partial sdd}(sx - dx) \\ r_{21} & r_{22} & r_{23} & -sy - \frac{\partial sid}{\partial sdd}(sy - dy) \\ 0 & 0 & 0 & -\frac{\partial sid}{\partial sdd} \end{bmatrix}$$

#### b- Dérivée par rapport à sid

$$\frac{\partial f}{\partial sid} = \begin{bmatrix} \frac{\partial sdd}{\partial sid} r_{11} & \frac{\partial sdd}{\partial sid} r_{12} & \frac{\partial sdd}{\partial sid} r_{13} & -\frac{\partial sdd}{\partial sid} sx - (sx - dx) \\ \frac{\partial sdd}{\partial sid} r_{21} & \frac{\partial sdd}{\partial sid} r_{22} & \frac{\partial sdd}{\partial sid} r_{23} & -\frac{\partial sdd}{\partial sid} sy - (sy - dy) \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

Au premier ordre, on peut supposer que  $sid$  et  $sdd$  sont proportionnel. En posant  $sdd = k \times sid$ . On obtient :

$$\frac{\partial f}{\partial sdd} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & -sx - \frac{1}{k}(sx - dx) \\ r_{21} & r_{22} & r_{23} & -sy - \frac{1}{k}(sy - dy) \\ 0 & 0 & 0 & -\frac{1}{k} \end{bmatrix} \\ \frac{\partial f}{\partial sid} = \begin{bmatrix} k r_{11} & k r_{12} & k r_{13} & -k sx - (sx - dx) \\ k r_{21} & k r_{22} & k r_{23} & -k sy - (sy - dy) \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

**c- Dérivée par rapport à dx**

$$\frac{\partial f}{\partial dx} = \begin{bmatrix} -r_{31} & -r_{32} & -r_{33} & -sid \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

**d- Dérivée par rapport à dy**

$$\frac{\partial f}{\partial dy} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ -r_{31} & -r_{32} & -r_{33} & -sid \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

**e- Dérivée par rapport à sx**

$$\frac{\partial f}{\partial sx} = \begin{bmatrix} r_{31} & r_{32} & r_{33} & sid - sdd \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

**f- Dérivée par rapport à sy**

$$\frac{\partial f}{\partial sy} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ r_{31} & r_{32} & r_{33} & sid - sdd \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

**3- Dérivée de la matrice de projection par rapport aux angles.**

**a- Formules**

Nous avons :

$${}^d_w \mathbf{R} = \begin{bmatrix} c(\theta)c(\psi) & c(\theta)s(\psi) & -s(\theta) \\ s(\varphi)s(\theta)c(\psi) - c(\varphi)s(\psi) & s(\varphi)s(\theta)s(\psi) + c(\varphi)c(\psi) & s(\varphi)c(\theta) \\ c(\varphi)s(\theta)c(\psi) + s(\varphi)s(\psi) & c(\varphi)s(\theta)s(\psi) - s(\varphi)c(\psi) & c(\varphi)c(\theta) \end{bmatrix}$$

$$\frac{\partial {}^d_w \mathbf{R}}{\partial \varphi} = \begin{bmatrix} 0 & 0 & 0 \\ c(\psi)c(\varphi)s(\theta) + s(\varphi)s(\psi) & s(\varphi)s(\theta)s(\psi) + c(\varphi)c(\psi) & s(\varphi)c(\theta) \\ -s(\varphi)c(\psi)s(\theta) + c(\varphi)s(\psi) & -s(\varphi)s(\theta)s(\psi) - c(\varphi)c(\psi) & -s(\varphi)c(\theta) \end{bmatrix}$$

$$\frac{\partial {}^d_w \mathbf{R}}{\partial \theta} = \begin{bmatrix} -c(\psi)s(\theta) & -s(\theta)s(\psi) & -c(\theta) \\ c(\psi)s(\varphi)c(\theta) & s(\varphi)s(\psi)c(\theta) & -s(\theta)s(\varphi) \\ c(\varphi)c(\psi)c(\theta) & c(\varphi)s(\psi)c(\theta) - s(\varphi)c(\psi) & -c(\varphi)s(\theta) \end{bmatrix}$$

$$\frac{\partial {}^d_w \mathbf{R}}{\partial \psi} = \begin{bmatrix} -c(\theta)s(\psi) & c(\theta)c(\psi) & 0 \\ -s(\varphi)s(\theta)s(\psi) - c(\varphi)c(\psi) & s(\varphi)s(\theta)c(\psi) - c(\varphi)s(\psi) & 0 \\ -c(\varphi)s(\theta)s(\psi) + s(\varphi)c(\psi) & c(\varphi)s(\theta)c(\psi) + s(\varphi)s(\psi) & 0 \end{bmatrix}$$

où :

$$\frac{\partial f}{\partial x} = \begin{bmatrix} sdd \frac{\partial r_{11}}{\partial x} + (sx - dx) \frac{\partial r_{31}}{\partial x} & sdd \frac{\partial r_{12}}{\partial x} + (sx - dx) \frac{\partial r_{32}}{\partial x} & sdd \frac{\partial r_{13}}{\partial x} + (sx - dx) \frac{\partial r_{33}}{\partial x} & 0 \\ sdd \frac{\partial r_{21}}{\partial x} + (sy - dy) \frac{\partial r_{31}}{\partial x} & sdd \frac{\partial r_{22}}{\partial x} + (sy - dy) \frac{\partial r_{32}}{\partial x} & sdd \frac{\partial r_{23}}{\partial x} + (sy - dy) \frac{\partial r_{33}}{\partial x} & 0 \\ \frac{\partial r_{31}}{\partial x} & \frac{\partial r_{32}}{\partial x} & \frac{\partial r_{33}}{\partial x} & 0 \end{bmatrix}$$

où  $x = \{\varphi, \theta, \psi\}$ , et :

$$\frac{\partial {}^{d'}\mathbf{R}}{\partial x} = \begin{bmatrix} \frac{\partial r_{11}}{\partial x} & \frac{\partial r_{12}}{\partial x} & \frac{\partial r_{13}}{\partial x} \\ \frac{\partial r_{21}}{\partial x} & \frac{\partial r_{22}}{\partial x} & \frac{\partial r_{23}}{\partial x} \\ \frac{\partial r_{31}}{\partial x} & \frac{\partial r_{32}}{\partial x} & \frac{\partial r_{33}}{\partial x} \end{bmatrix}$$

### b- Approximation petits angles

Dans le cas où seule la rotation autour de  ${}^d\mathbf{y}$  est prise en compte. Les angles  $\varphi$  et  $\psi$  peuvent être considérés comme suffisamment petits pour être approximés via une approximation petits angles. Pour rappel, dans le cadre d'une approximation petits angles on peut considérer que si  $x$  est un angle suffisamment petit :  $\cos(x) = 1$  et  $\sin(x) = x$  (premier terme du développement limité en 0 des fonctions cosinus et sinus). On obtient :

$$\frac{\partial {}^{d'}\mathbf{R}}{\partial \varphi} = \begin{bmatrix} 0 & 0 & 0 \\ s(\theta) + \varphi\psi & \varphi\psi s(\theta) + 1 & \varphi c(\theta) \\ -\varphi s(\theta) + \psi & -\varphi\psi s(\theta) - 1 & -\varphi c(\theta) \end{bmatrix}$$

$$\frac{\partial {}^{d'}\mathbf{R}}{\partial \theta} = \begin{bmatrix} -s(\theta) & -\psi s(\theta) & -c(\theta) \\ \varphi c(\theta) & \varphi\psi c(\theta) & -\varphi s(\theta) \\ c(\theta) & \psi c(\theta) - \varphi & -s(\theta) \end{bmatrix}$$

$$\frac{\partial {}^{d'}\mathbf{R}}{\partial \psi} = \begin{bmatrix} -c(\theta)\psi & c(\theta) & 0 \\ -\varphi s(\theta)\psi - 1 & \varphi s(\theta) - \psi & 0 \\ -s(\theta)\psi + \varphi & s(\theta) + \varphi\psi & 0 \end{bmatrix}$$

## 4- Influence des différents paramètres sur le projeté 2D :

*Si non précisées, les valeurs sont données en **rad** pour les angles et **m** pour les distances.*

Afin de visualiser l'influence des différents paramètres sur les projetés 2D. On simule le scénario suivant : Nous projetons (grâce à la matrice P) une grille de 5 x 5 marqueurs sur le détecteur plan. Nous choisissons une distance source-détecteur :  $sdd=1.5\text{m}$  et une distance source-point de reconstruction de  $swd=0.7\text{m}$ . Nous supposons que la grille est centrée sur l'origine du plan de reconstruction et orientée de façon à être parallèle au plan détecteur comme montré sur la Fig.9.

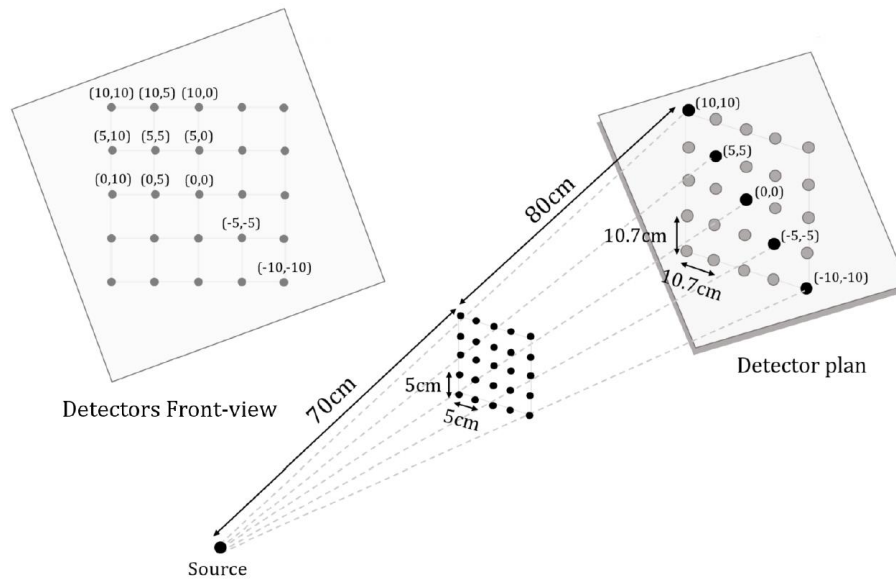


Fig.9 : Scénario de la propagation d'erreur jusqu'au projeté 2D

Nous allons étudier par la suite l'influence de ces valeurs sur l'impact de certains paramètres. Pour la visualisation de l'impact des erreurs sur la grille projetée, nous simulons des valeurs d'erreurs comparables sur les différents paramètres à savoir :

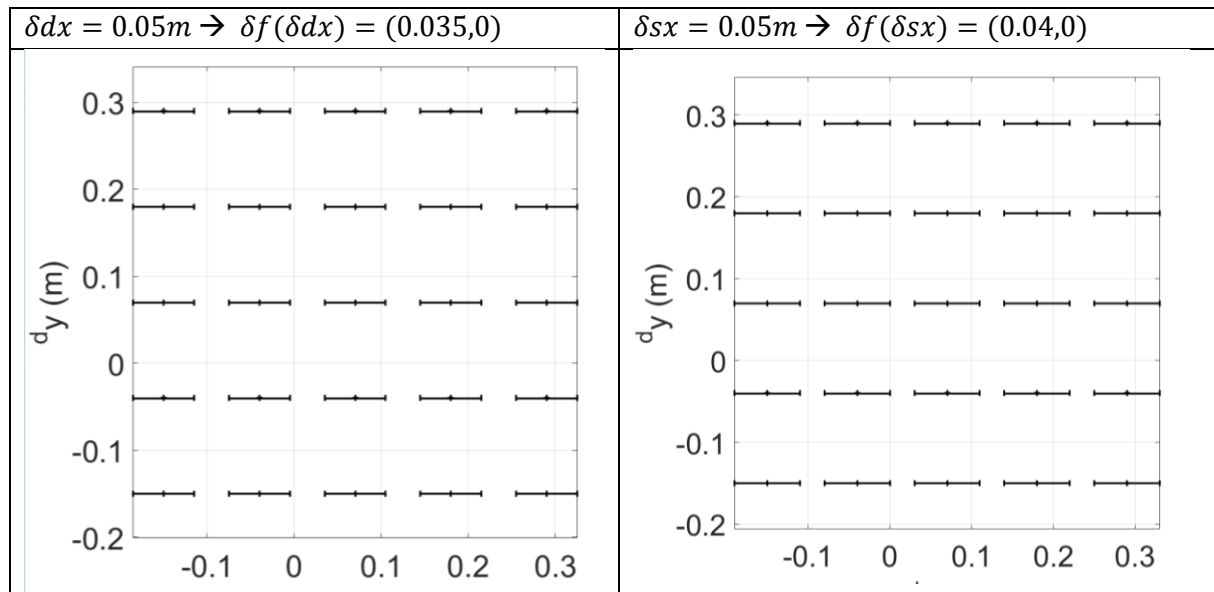
- $\delta \varphi = \delta \theta = \delta \psi = 0.075 \text{ rad} (\approx 4.3^\circ)$
- $\delta s_x = \delta s_y = \delta d_x = \delta d_y = \delta s_{dd} = \delta s_{id} = 0.05 \text{ m}$ .

Les valeurs choisies sont sciemment choisies assez importantes afin de faciliter la visualisation de l'impact de l'erreur sur les différents paramètres. Nous n'allons pas utiliser de convention petits angles dans cette simulation. Une seule valeur d'erreur est prise en compte à chaque simulation les autres  $\delta x$  sont mis à 0.

### a- Erreur selon dx et sx

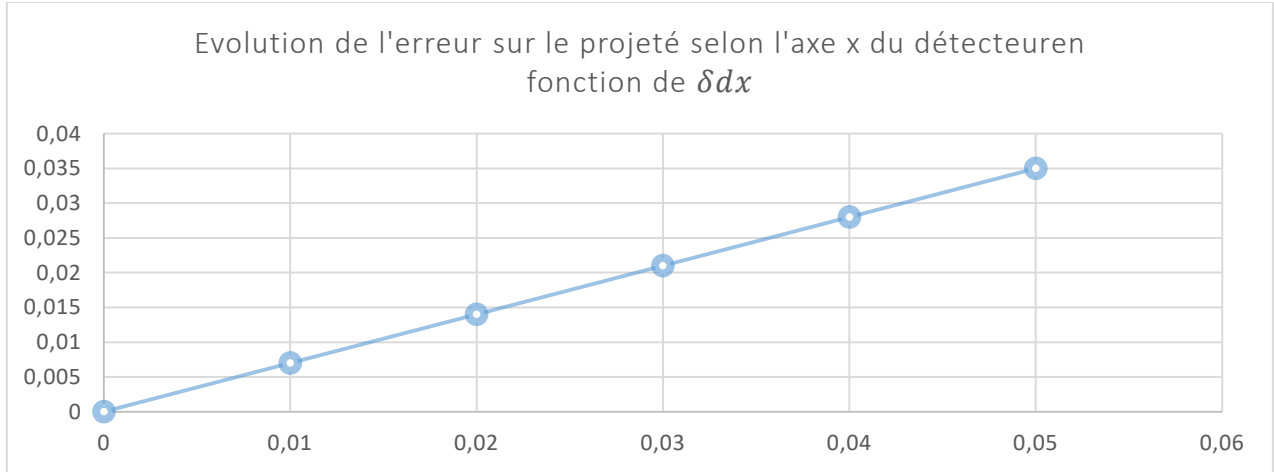
#### • Visualisation sur la grille de projection

Une erreur selon dx et sx donne une erreur horizontale



Remarque : Nous retrouvons bien  $\frac{\delta f(\delta dx)}{\delta f(\delta sx)} = \frac{35}{40} = \frac{swd}{sdd-swd}$

- **Evolution**

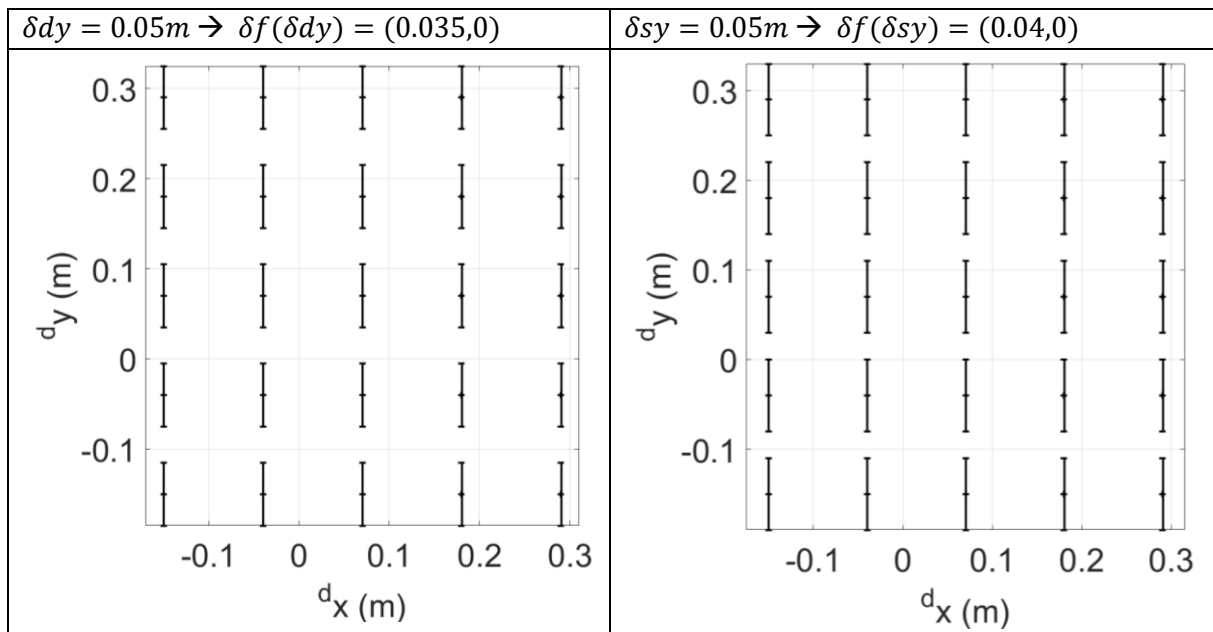


Comme on peut le voir l'évolution de l'erreur selon  $dx$  (analogue à  $sx$ ) est linéaire.

**b- Erreur selon  $dy$  et  $sy$**

- **Visualisation sur la grille de projection**

Une erreur selon  $dx$  et  $sx$  donne une erreur verticale



Remarque : De même, nous retrouvons bien  $\frac{\delta f(\delta dy)}{\delta f(\delta sy)} = \frac{35}{40} = \frac{swd}{sdd-swd}$

- **Evolution**

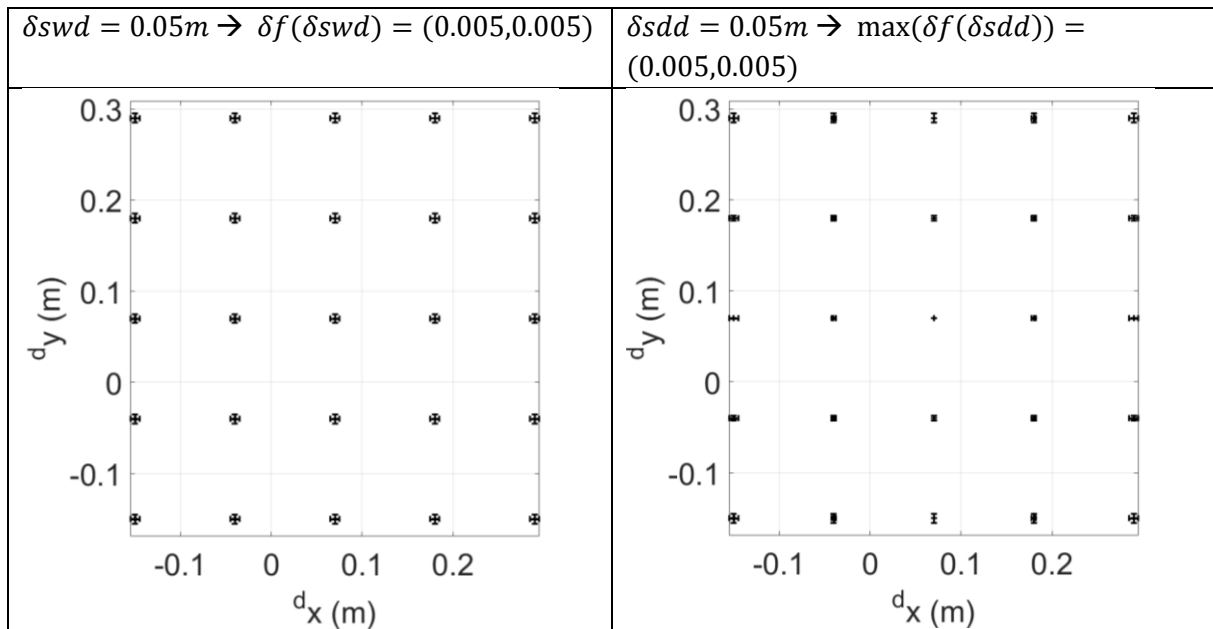
De façon analogue à  $dx$  et  $sx$ , L'erreur sur la matrice de projection évolue de façon linéaire en fonction de  $dy$  et  $sy$ .



### c- Erreur selon swd et sdd

- **Visualisation sur la grille de projection :**

1. Une erreur selon *swd* entraîne une erreur uniforme sur la totalité de la grille de même valeur selon  $d_x$  et  $d_y$ .
2. Une erreur selon *sdd* entraîne une erreur non uniforme sur la totalité de la grille. Elle est plus important sur les bords et nulle au milieu de la grille. L'évolution de l'erreur pour différents points de la grille et selon la valeur de l'erreur est présentée sur la **courbe X**.
3. Les valeurs selon *swd* et *sdd* sont globalement comparables (valeurs égales aux bords)
4. Globalement une erreur de même valeur sur la translation hors-plan : *swd* et *sdd* impacte moins (ici un rapport 8) le projeté qu'une erreur sur les translations dans le plan  $s_x, s_y, dx$  et  $dy$ .



Note : Dans cet exemple, les erreurs sont les suivantes pour les erreurs entraînées par  $\delta sdd = 0.05m$  sur différents point de la grille **figure x** :

point	(0 ; 0)	(0 ; 5)	(0 ; 10)	(5 ; 0)	(5 ; 5)	(5 ; 10)	(10 ; 0)	(10 ; 5)	(10 ; 10)
erreur	(0 ; 0)	(0.0025 ; 0)	(0.005 ; 0)	(0 ; 0.0025)	(0.0025 ; 0.0025)	(0.005 ; 0.0025)	(0 ; 0.005)	(0.0025 ; 0.005)	(0.005 ; 0.005)

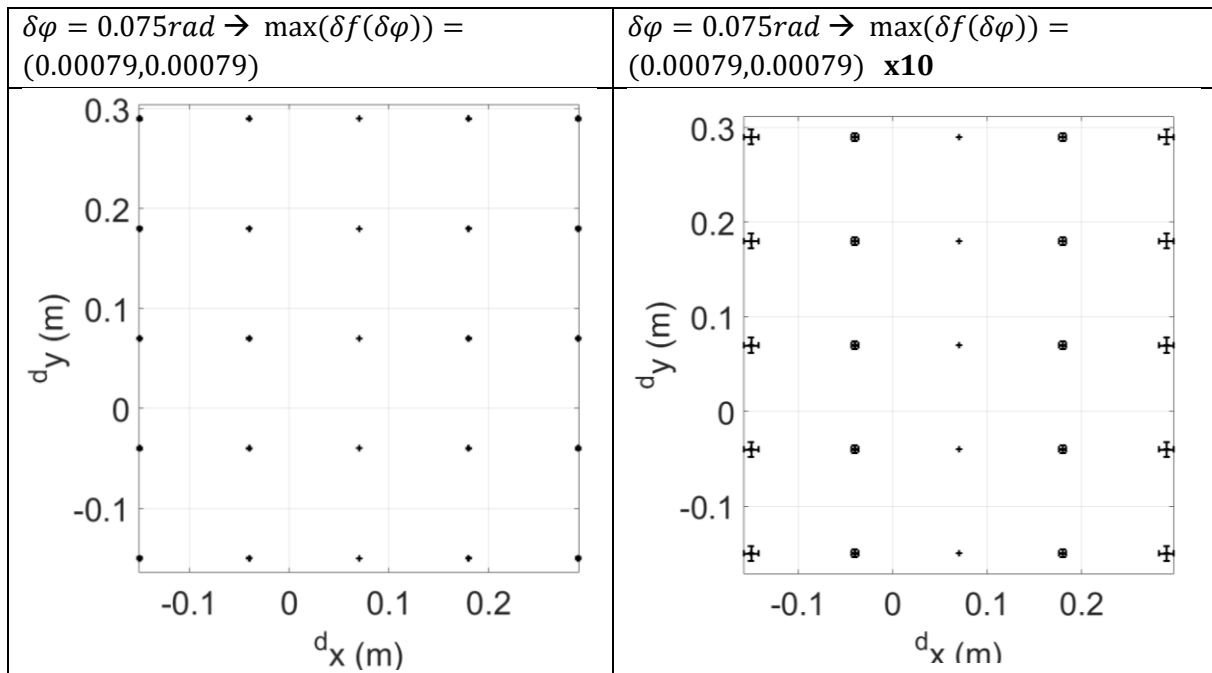
- **Evolution de l'erreur**

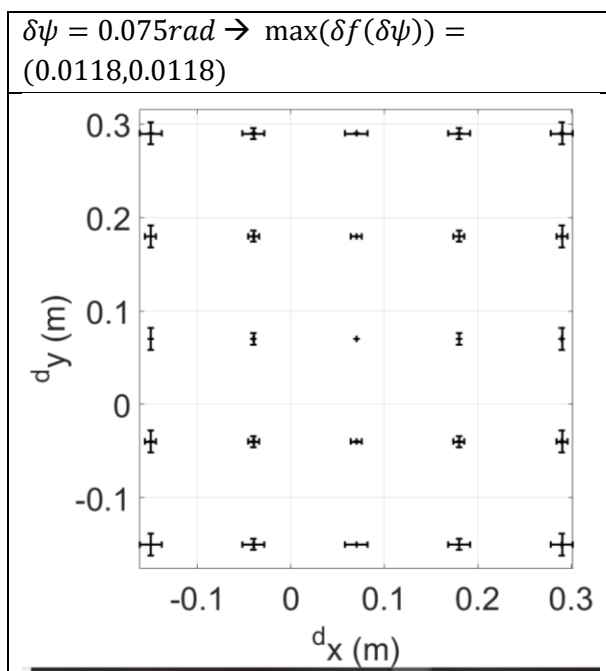
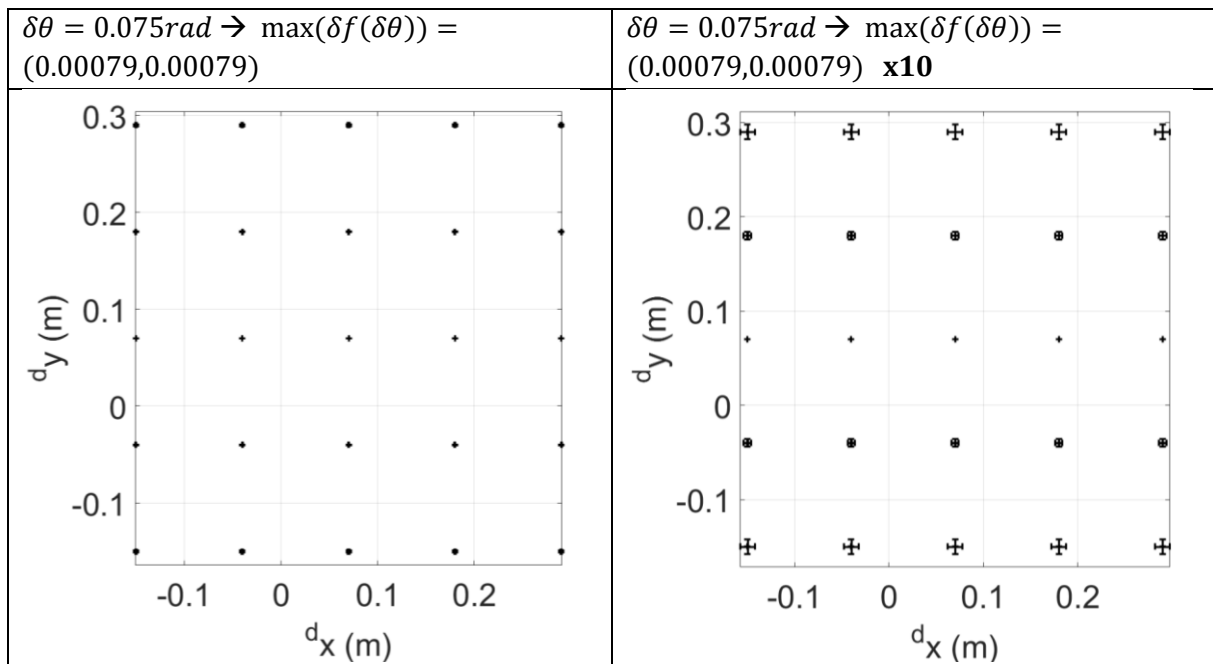
De même, l'évolution de l'erreur sur le projeté entraînées par  $\delta swd$  et  $\delta sdd$  évolue de façon linéaire en fonction de l'erreur sur les paramètres.

#### d- Erreur selon $\varphi$ , $\theta$ et $\psi$

- **Visualisation sur la grille de projection :**

1. Les erreurs sur les angles sont en rad. afin de pouvoir comparer l'erreur entraînée par les erreurs sur les angles et celles entraînées par les erreurs sur les distances, en remarquant que les angles sont suffisamment petits nous posons que  $erreur\ en\ rad = \frac{erreur\ en\ m}{sdd}$  (approximation petit angle et expression du sinus).
2. Comme les erreurs sur les projetés entraînées par les angles  $\varphi$  et  $\theta$  sont petites, nous allons afficher l'erreur  $\times 10$  à droite de la vraie grille obtenue pour chaque angle.
3. Comme nous pouvons le voir, les 3 angles n'affectent pas de façon uniforme l'ensemble de la grille. Les erreurs entraînées par  $\delta\varphi$  présentent une symétrie autour d'un axe vertical centrée en  $dy-sy$ . Celles entraînées par  $\delta\theta$  présentent une symétrie autour d'un axe horizontal centré en  $dx-sx$ . Les erreurs entraînées par  $\delta\psi$ , elles sont symétriques par rapport au point  $(dx-sx, dy-sy)$ .
4. Globalement, les erreurs entraînées par les angles influent moins que les erreurs sur les translations dans le plan.  $\psi$  est l'angle qui impacte le plus le projeté.





- **Evolution de l'erreur**

De même, l'évolution de l'erreur sur le projeté entrainées par  $\delta\varphi$ ,  $\delta\theta$  et  $\delta\psi$  évolue de façon linéaire en fonction de l'erreur sur chacun des paramètres.

**Remarque : Influence de  $|s_x - dx|$**

## VIII- Bras rotatif

Le bras rotatif est notre première plateforme expérimentale. Elle représente la source ou le détecteur d'un C-arm simplifié. C'est une table tournante motorisée grâce à un moteur brushless et équipée d'un bras en bois. Nous avons monté des centrales à inertie de type MPU6050 et Navchip à différents endroits de cette dernière (ANNEXE 1).

- Les dimensions de la table tournante sont présentées en Annexe 1,
- La Fig.10 présente un schéma simplifié ainsi que des photographies de la table tournante.

### 1- Présentation du bras rotatif :

Le bras rotatif se présente comme qui suit :

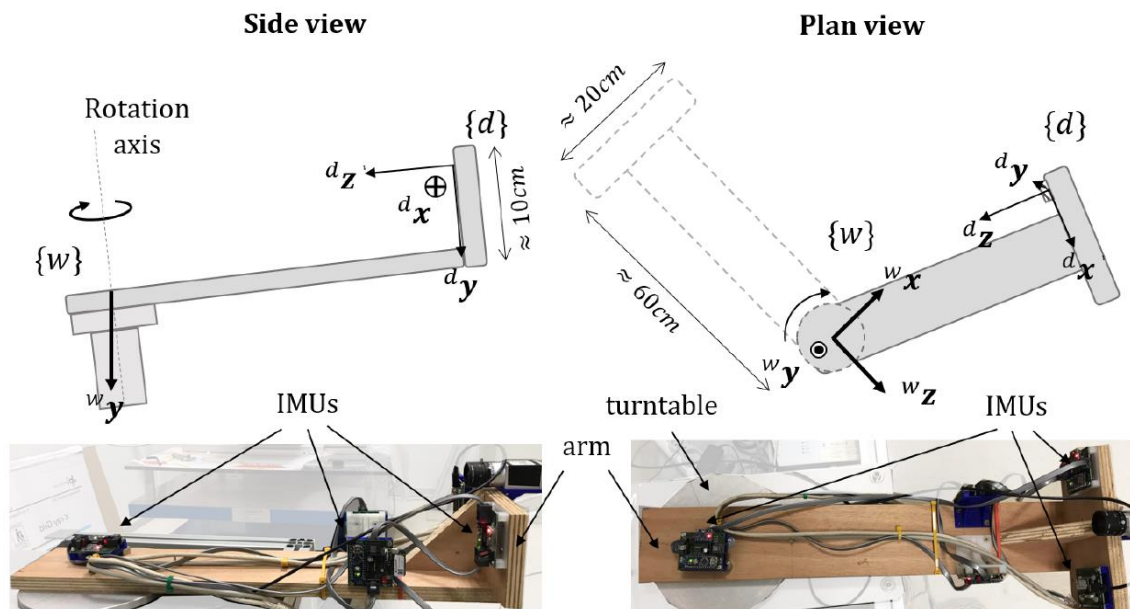


Fig.10 : Le bras tournant. Schémas (haut) et photos (bas). Reproduit le mouvement du détecteur ou de la source d'un C-arm

Comme dit plus haut, la table tournante peut représenter la source ou le détecteur d'un C-arm. C'est un système isocentrique dont l'axe de rotation est fixe et légèrement incliné par rapport au vecteur de gravité terrestre. Comme nous allons le voir, la table tournante est sujette à d'importantes vibrations mécaniques et présente une très bonne répétabilité de mouvement (99% dans toutes les directions). La bonne connaissance à priori qu'on a sur la table tournante nous a permis de construire un simulateur réaliste des trajectoires réalisées par ces dernières. Le simulateur va nous donner une « réalité » simulée à laquelle nous pourrions comparer nos positions et orientation estimées afin d'évaluer leur précision.

### 2- Stratégies de mouvement du bras rotatif

Différents scénarios sont configurables dont les paramètres sont les suivants (extrait de « Manager et stratégies d'acquisition ») :

1. **Rotation continu** (*ContinuousStrategy*) : À partir d'une position initiale, la table tournante effectue un mouvement de rotation continu défini par l'utilisateur. Tous les "pos\_increment" degrés, elle envoie un signal matériel aux IMU synchronisant ainsi leurs acquisitions.

#### Paramètres

- length : nombre de pulses générées pendant le mouvement
- start\_pos : Position en degré de démarrage de la stratégie
- pos\_increment : step angulaire (en °) / pulse
- speed : vitesse de rotation du moteur en °/seconde.
- round\_trip : false, le mouvement est un arc de cercle simple ; true il fait un "aller-retour\_avec\_dépassement-aller"

2. **Angle constant** (*MoveAndShootStrategy*) : Après avoir positionné la table à un certain angle, cette dernière reste fixe. Les acquisitions des IMU sont synchronisées au moyen d'un signal matériel dit *interne*. Le nombre d'acquisitions est fixé par l'utilisateur

#### Paramètres

- nb\_shoots : nombre de pulses générées dès que la stratégie est lancée (= nb de shoot image et trig IMU)
- start\_pos : Position en degré de démarrage de la stratégie Vs la position zéro
- speed : vitesse de rotation du moteur pour aller à start\_pos en °/seconde.
- sync\_period : Period en µsec entre chaque pulses

3. **Angle constant** (*StepAndShootStrategy*) : Après avoir positionné la table à un certain angle (start\_pos), la table tournante effectue un mouvement de rotation par step de "pos\_increment" degrés. À chaque step la table s'arrête, fait un certains nombre "length" d'acquisitions. Le nombre de steps et le nombre de pulses sont fixés par l'utilisateur.

#### Paramètres

- length : nombre de step angulaire total de la stratégie
- nb\_shoots : nombre de pulses générées dès que la turntable a effectué un step angulaire (= n b de shoot image et trig IMU)
- start\_pos : Position en degré de démarrage de la stratégie Vs la position zéro
- pos\_increment : valeur en degré du step angulaire
- speed : vitesse de rotation du moteur pour aller à start\_pos en °/seconde.
- sync\_period : Period en µsec entre chaque pulses

4. **Mouvement libre** (*FreeMoveStrategy*) : Le mouvement est laissé entièrement libre à l'utilisateur (aucune table tournante n'est utilisée). Les périphériques d'acquisition peuvent être de type "hardware" impliquant un module de synchronisation matériel Si, à l'inverse, les périphériques sont émulés, la synchronisation est logicielle. Il n'est pas possible de mélanger des extensions matérielles et émulées. Le nombre d'acquisitions est défini par l'utilisateur.

#### Paramètres

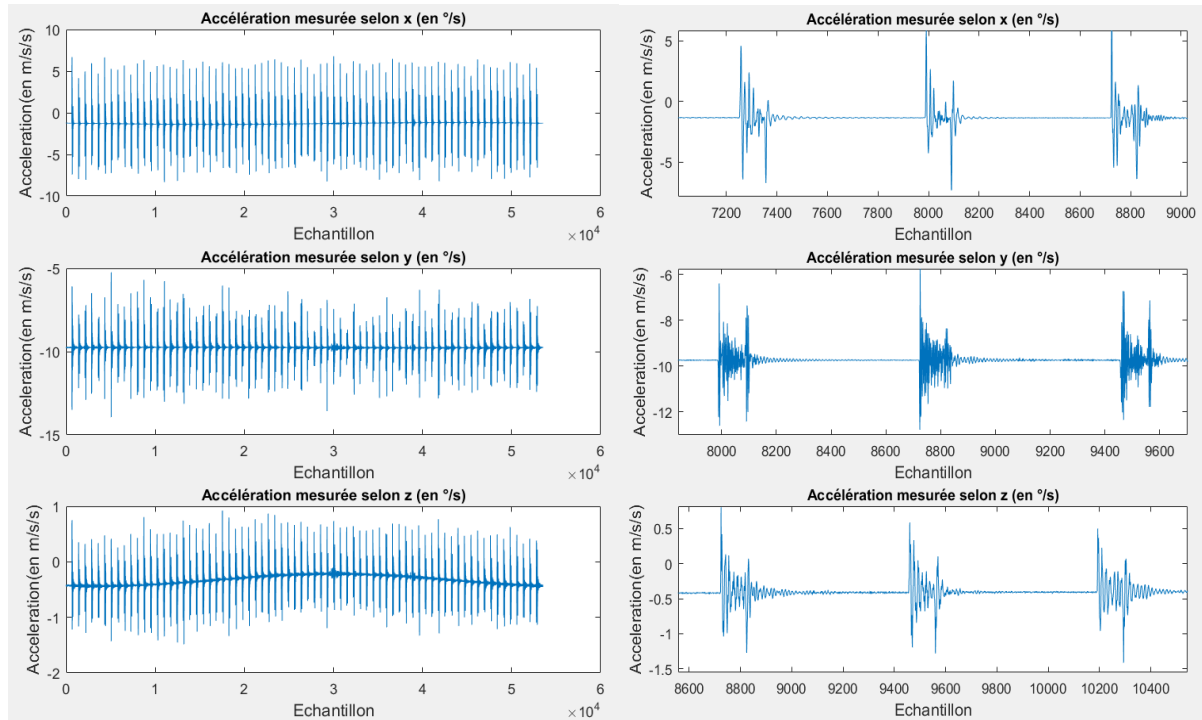
- nb\_shoots : nombre de pulses générées dès que la stratégie est lancée (= nb de shoot image et trig IMU)
- sync\_period : Period en µsec entre chaque pulses

### 3- Caractérisation de la table tournante

Dans cette sous partie, je vais relater le processus que j'ai suivi pour caractériser la table tournante en exploitant des acquisitions statiques du ISCN02.

- **Etape 1** : Acquisition

Pour la caractérisation nous effectuons une rotation de **360°** à une vitesse **10°/s** en **Step and Shoot** avec des intervalles de **5°** (**72** intervalles de **1** à **73**), durée de chaque pallier de **3s** et Fréquence d'échantillonnage de **200Hz**. Les courbes des acquisitions de l'accéléromètre 3 axes sont affichées *Fig. 11*.



*Fig 11: Mesures de l'accéléromètre 3 axes. 360° à 10°/s en Step and Shoot avec des intervalles de 5° et des arrêts de 3°. (Gauche : signal global, Droite : zoom sur 3 steps du signal)*

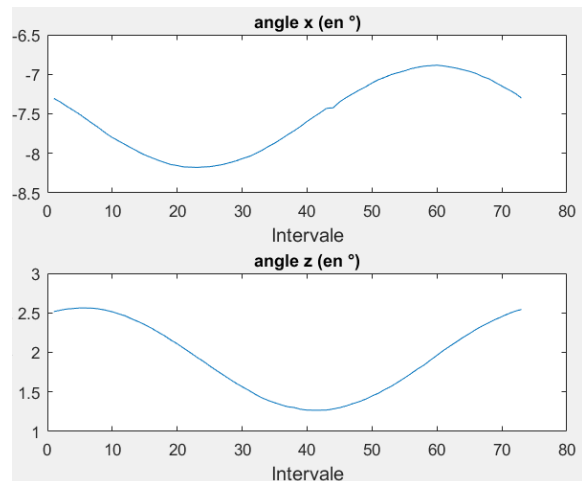
- **Etape 2** : Extraire les angles des mesures de l'accéléromètre

On calcule la moyenne de **200 échantillons** de chaque pallier après **stabilisation du signal**. On calcule  $angle_x$  et  $angle_z$  les angles autour de  $x$  et  $z$  respectivement selon les formules :

$$angle_x = \arcsin\left(\frac{acc_{moyenne_x}}{g}\right)$$

$$angle_z = \arctan\left(\frac{acc_{moyenne_z}}{acc_{moyenne_y}}\right)$$

*Fig 12 : Angles extraits des mesures de l'accéléromètre autour des axes x et z*



## Remarques

- Qualitativement, le mouvement a une allure sinusoïdale cohérente
- Même valeur pour l'intervalle 1 et 73 ( $angle_x$  : -7.305 pour intervalle=1 et -7.3 pour intervalle=73 ;  $angle_z$ : 2.516 pour intervalle=1 et 2.544 pour intervalle=73 – Résultat moins bon pour psi... pourquoi ?)
- Le mouvement est bien reproductible
- Le saut observé au niveau de l'intervalle 45 sur cette acquisition n'est pas reproductible et reste inexpliqué.

- **Etape 3** : Calcul de l'inclinaison de la table

Prenons pour exemple l'inclinaison autour de l'axe  $x_t$

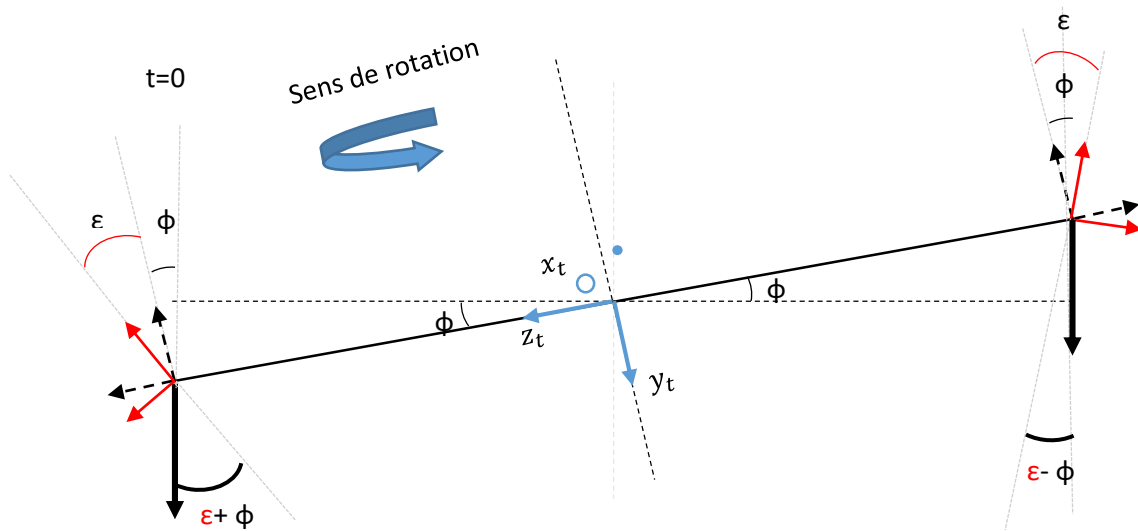


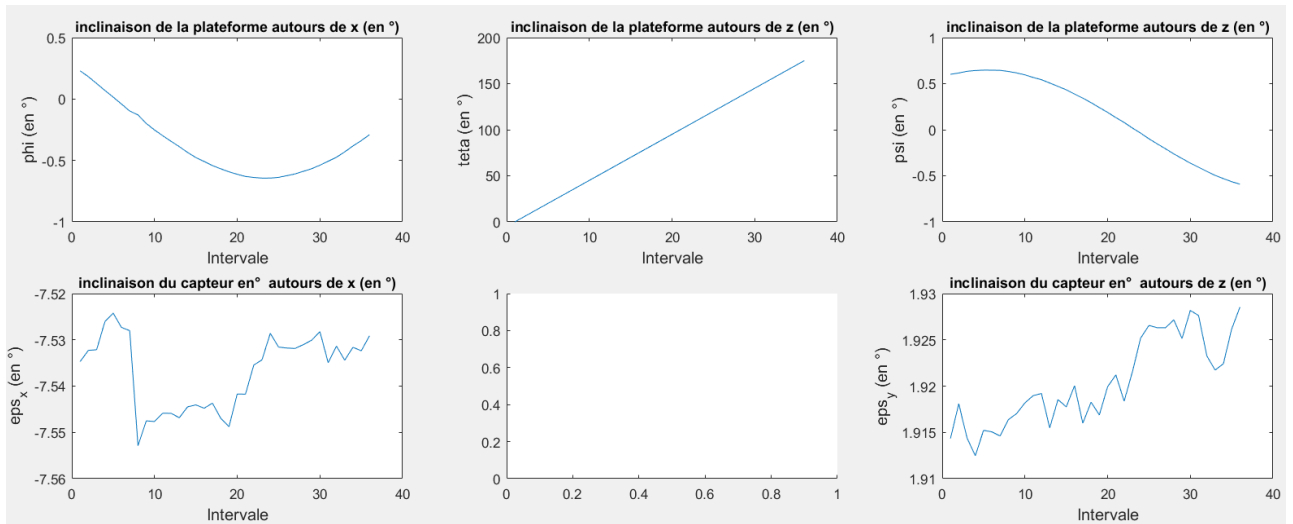
Fig 13 : Schéma explicatif des formules de caractérisation de la table tournante

On suppose qu'on connaît précisément  $\theta$  l'angle de rotation autour de l'axe normal à la table  $y_t$  tournante. Donc :

$$\varepsilon_{mes}(\theta^\circ) = -\varepsilon_{mes}(\theta^\circ + 180^\circ) = \frac{angle_x(\theta^\circ) - angle_x(\theta^\circ + 180^\circ)}{2} \text{ (Inclinaison du capteur)}$$

$$\varphi_{mes}(\theta^\circ) = -\varphi_{mes}(\theta^\circ + 180^\circ) = \frac{angle_x(\theta^\circ) + angle_x(\theta^\circ + 180^\circ)}{2} \text{ (Inclinaison de la table tournante)}$$

On suit le même résultat pour  $\psi_{mes}$  l'inclinaison autour de l'axe  $z_t$ .



### Remarques :

- Nous retrouvons bien une inclinaison du capteur constante au  $0.02^\circ$  près (un ordre de grandeur entre l'inclinaison du capteur et l'inclinaison de la table tournante).
- Les résultats sont améliorables en faisant un tour avec de plus petits pas et une moyenne de plusieurs acquisitions

### Exploitation qualitative :

- La valeur  $\varphi_{acc}=0.01569^\circ$  (la plus proche de  $0^\circ$ ) correspond à  $\theta_{acc}=20^\circ$  et  $\psi_{acc}=0.6468^\circ$  (extrémum).
- La valeur  $\psi_{acc}=0.01814^\circ$  (la plus proche de  $0^\circ$ ) correspond à  $\theta_{acc}=110^\circ$  et  $\varphi_{acc}=-0.6438^\circ$  (extrémum).
- Comme on peut le remarquer l'extremum de  $\psi_{acc}$  correspond au 0 de  $\varphi_{acc}$  et vice-versa
- Les valeurs des extremums sont les mêmes et arrivent à un intervalle de  $\theta_{acc}=90^\circ$   
Nous pouvons dors déjà avoir l'intuition d'un axe de rotation incliné fixe qu'on note u de  $0.64^\circ$  selon z et  $0^\circ$  selon y pour une valeur de  $\theta_{acc}=20^\circ$ .

### Etape 4 : Matrice de rotation de la table autour de l'axe u

Le but est de prouver que la rotation de la table tournante s'effectue autour d'un axe fixe évoqué par avant. Pour cela on choisit un intervalle de travail par exemple **intervalle = 6**.

$$\varphi_{inter=6} = -0.0405^\circ$$

$$\theta_{inter=6} = -25^\circ$$

$$\psi_{inter=6} = 0.6464^\circ$$

On construit la matrice de rotation de ces 3 angles  $R_{inter=6}$  et on la multiplie par le vecteur de rotation orthogonal  $[0;1;0]$  pour obtenir l'axe de rotation pour  $\theta_{inter=6} = -25^\circ$ .



$$u = R_{inter=6} \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$$

On calcule  $R_u$  la matrice de rotation de la table tournante d'un angle  $\theta$  autour de  $u = \begin{pmatrix} a \\ b \\ c \end{pmatrix}$  selon l'expression :

$$R_u = \begin{pmatrix} 0 & -c & b \\ c & 0 & -a \\ -b & a & 0 \end{pmatrix} \sin(\theta) + (I_{3 \times 3} - uu^t) \cos(\theta) + uu^t$$

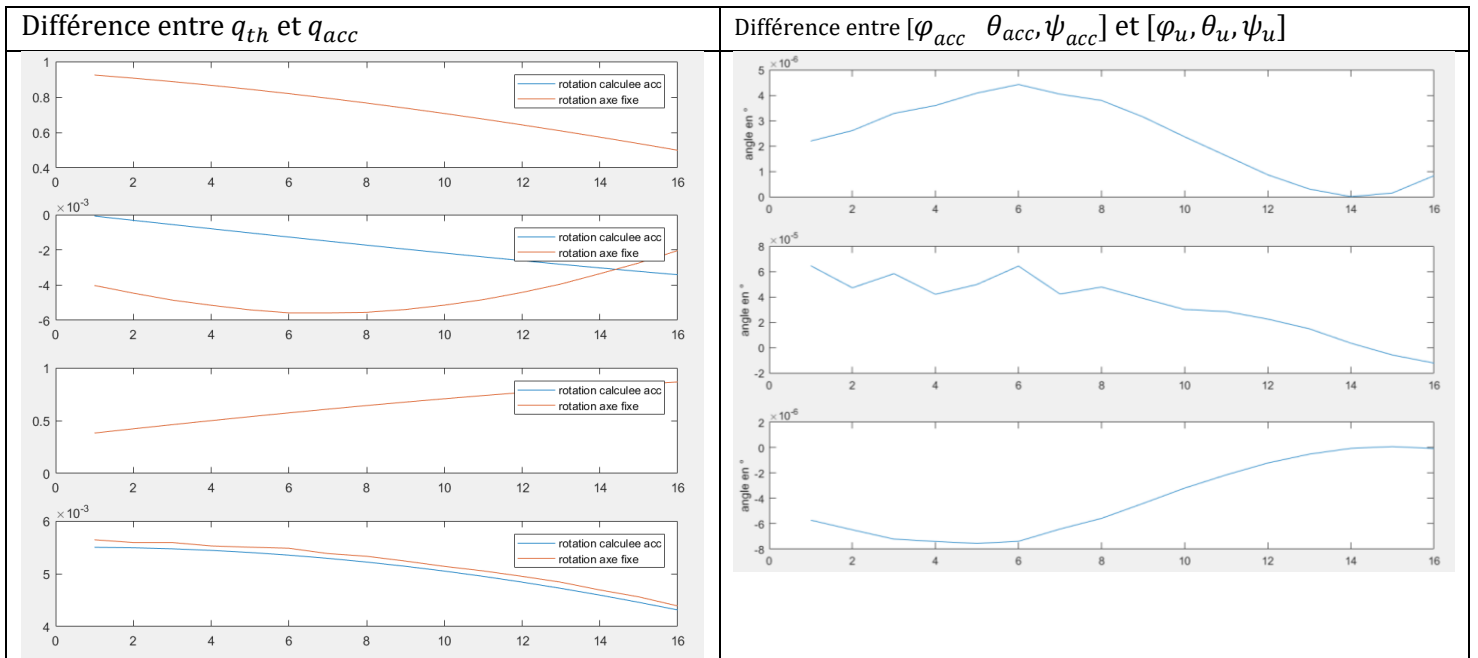
- **Etape 5** : Vérifions que u est un axe fixe

On compare les angles calculés à partir des mesures de l'accéléromètre ( $\varphi_{acc}, \theta_{acc}, \psi_{acc}$ ) et ceux obtenus via la rotation autour de l'axe fixe u ( $\varphi_u, \theta_u, \psi_u$ ). Si ces dernières correspondent, nous avons bien une rotation autour de l'axe u

La matrice de rotation  $R_{th}$  issue de la rotation autour de l'axe fixe u par rapport au repère monde s'écrit :

$$R_{th} = R_{inter=6} R_u$$

On extrait les quaternions  $q_{th}(R_{th})$  qu'on compare à  $q_{acc}(\varphi_{acc}, \theta_{acc}, \psi_{acc})$ . Dans les figures 4 et 5 on compare l'orientation pour  $q_{th}$  (intervalle 5 à 20)  $q_{acc}$  (intervalle 10 à 25), ce décalage est dû au fait que notre premier intervalle pour  $q_{th}$  est l'intervalle 6. Nous obtenons la courbe fig 4 pour la différence entre l'orientation représentée en quaternions **fig X** et en angles d'Euler **fig Y**



La différence entre  $(\varphi_{acc}, \theta_{acc}, \psi_{acc})$  et  $(\varphi_u, \theta_u, \psi_u)$  est de l'ordre maximum de  $10^{-5}$  degrés, nous pouvons conclure que le bras rotatif tourne autour d'un axe  $\vec{u}$  fixe, incliné d'angles  $(0.01569^\circ, 20^\circ, 0.6468^\circ)$  par rapport au vecteur de gravité  $\vec{g}$ .

#### 4- Répétabilité de la table tournante

##### - Etude qualitative :

Pour illustrer la répétabilité de la table tournante nous réalisons 2 rotations de  $360^\circ$  à une vitesse  $10^\circ/\text{s}$  en **Step and Shoot** avec des intervalles de  $5^\circ$  (72 intervalles de 1 à 73), durée de chaque palier de 3s et Fréquence d'échantillonnage de **200Hz** à partir de la position initiale. Les courbes de l'acquisition totale sont affichées sur la Fig.14 et un zoom est affiché sur la Fig.15.

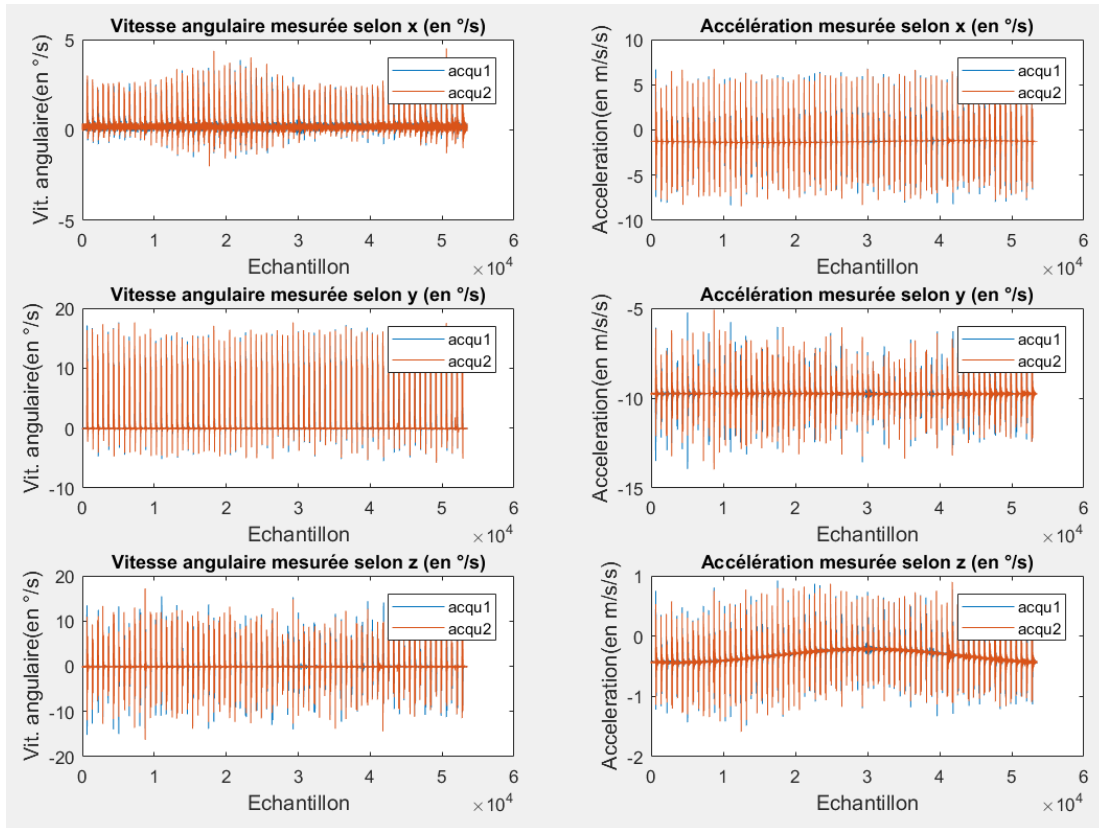


Fig 14 : Acquisition pour la répétabilité de la table tournante

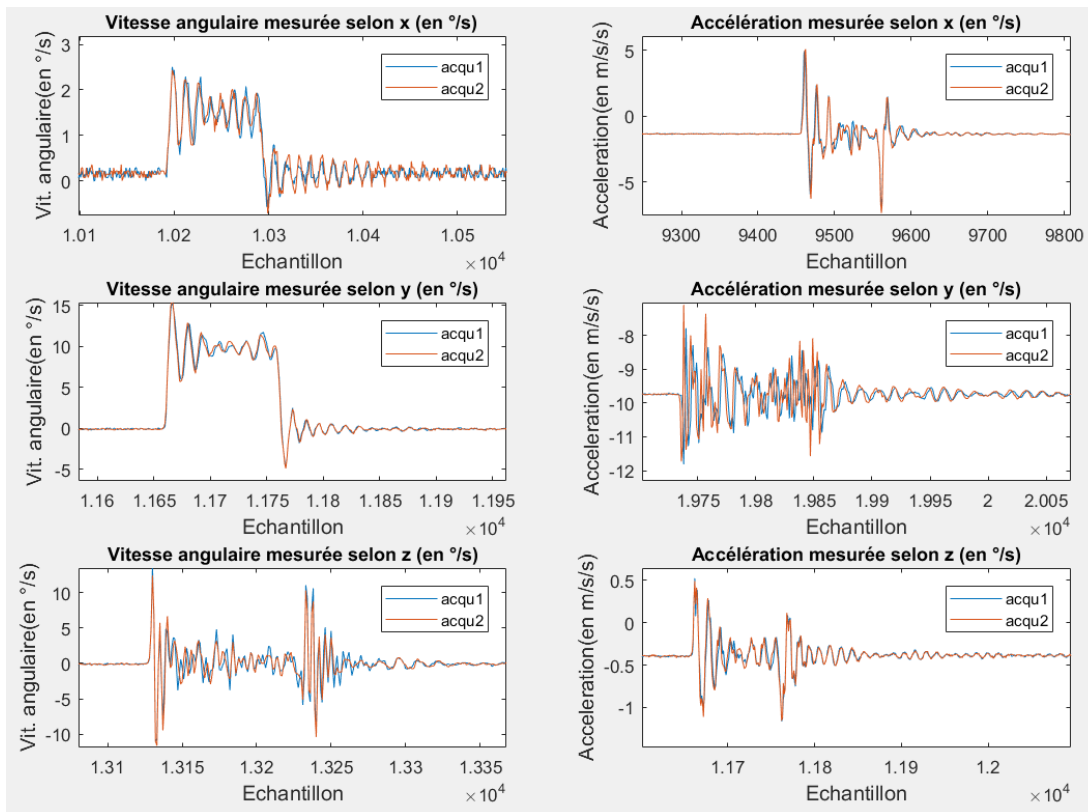


Fig 15 : Zoom, acquisition pour la répétabilité de la table tournante

On peut remarquer que les courbes des 2 acquisitions se superposent bien et que les pulsations mécaniques sont répétables on peut supposer qu'elles sont essentiellement issues de la réponse impulsionnelle de la table tournante.

**- Etude quantitative :**

Nous allons afficher les écarts types de mesures obtenus pour l'estimation de l'angle d'inclinaison de la table tournante à partir de 3 acquisitions. Deux acquisitions ont été réalisées à la suite et la troisième le lendemain matin à partir de la même position initiale.

intervalle	Ecart type $\varphi$ (en °)	Ecart type $\psi$ (en °)
1 (0° 180°)	.0056	.0054
2 (5° 185°)	.0074	.0019
3 (10° 190°)	.0030	.0026
4 (15° 195°)	.0068	.0019

Tab 4 : Ecart type de mesures entre 3 acquisitions statiques de la table tournante

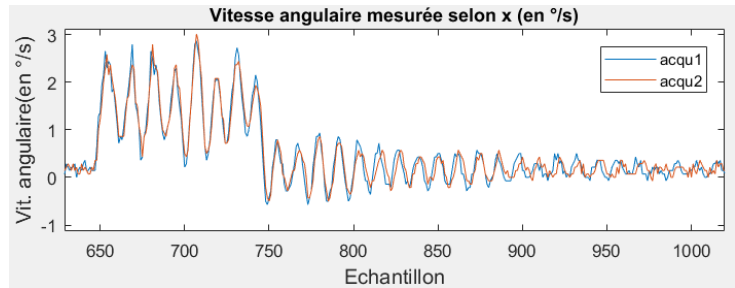
Les écarts types sont de l'ordre du millième de degré.

**Le bras rotatif est un système répétable.**

## 5- Petite discussion autour des pulsations

Nous zoomons sur les pulsations obtenues pour nos acquisitions :

Grosso modo on a 8 pulsations pour 100 échantillons et une fréquence de 200Hz donc 8 pulsations toutes les 0.5s. 1 pulsation dure 0.0625s donc notre système vibre à une fréquence de 16Hz. Les pulsations s'amortissent en environ 300 échantillons pour une acquisition à **166,6Hz**. (on retrouve un raie à 16Hz et harmonique lors de l'étude fréquentielle)



**J'affiche l'étude fréquentielle ou non sur ce rapport ?**

## 6- Simulateur

Pour pouvoir vérifier le bon fonctionnement des différents blocs et de mes modèles, j'ai implémenté un simulateur dont le schéma simplifié est présenté dans le schéma Fig.16. Il a trois fonctions principales :

- Tester le bon fonctionnement des algorithmes implémentés sur des trajectoires simples et maîtrisées (rotations et translations avec des erreurs de mesure connues).
- Prédire les limites des algorithmes et des modèles (ordre de discrétisation, modèle dynamique, décrochage...).
- Représenter une base de comparaisons pour les données expérimentales en prédisant le comportement idéal des systèmes de mesure.

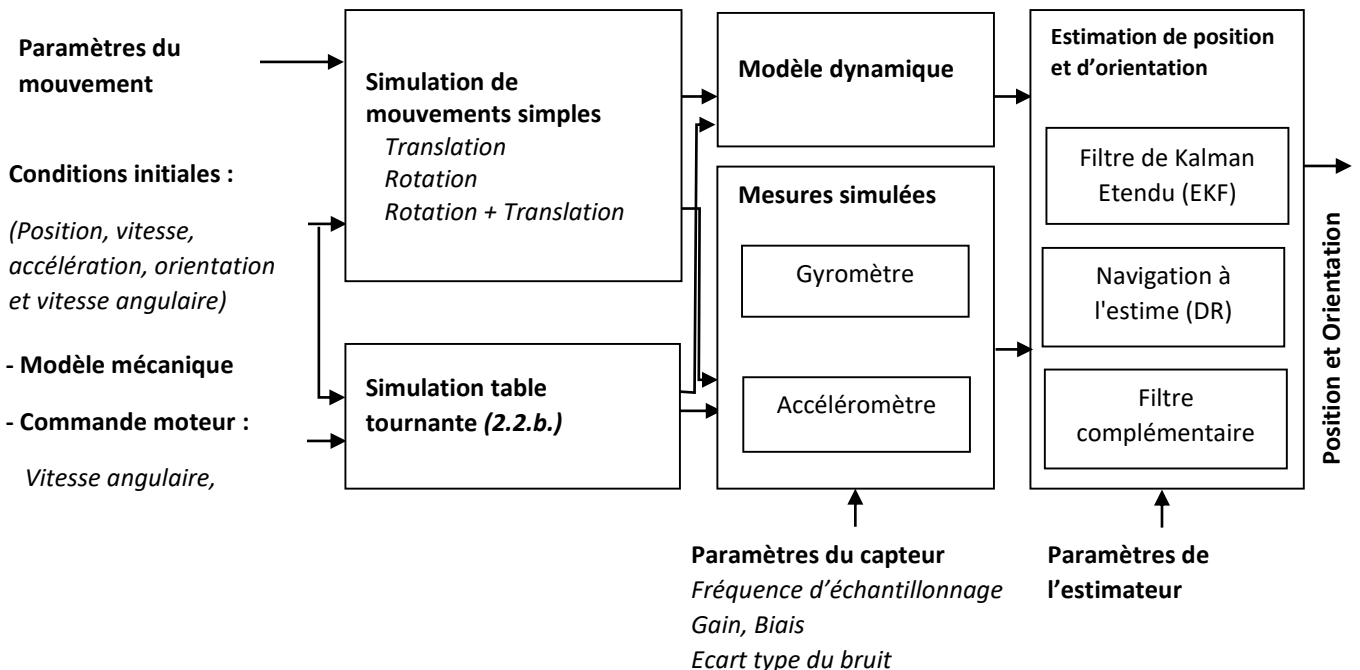


Fig.16 : Schéma simplifié de l'architecture du simulateur qui comprend la simulation de mouvements, l'acquisition des données par des modèles de capteurs et l'estimation de la position et de l'orientation

## **7- Conclusion : de la table tournante au C-arm**

Pour rappel, l'arceau chirurgical mobile (C-arm) est léger, instable mécaniquement et le plus souvent non motorisé. Il est sensible à l'attraction gravitationnelle qui éloigne la trajectoire du couple source détecteur d'un mouvement circulaire. Les trajectoires sont alors, non iso centriques. Le C-arm ne dispose, également, pas de l'amplitude de mouvement a priori nécessaire pour un scan tomographique ( $360^\circ$  par définition du scanner et  $200^\circ$  au minimum).

La plateforme expérimentale représente le détecteur/ la source rayons X d'un C-arm simplifié, motorisé, iso-centrique et dont la trajectoire est reproductible. Néanmoins la trajectoire s'éloigne d'un mouvement circulaire autour d'un axe fixe et ce dernier est le siège de vibrations mécaniques. Le bras rotatif dispose d'une meilleure répétabilité de mouvement et, sur lequel nous disposons d'une meilleure connaissance a priori. Elle va nous permettre d'effectuer un premier test de nos algorithmes de localisation et d'obtenir de premières estimations d'erreur sur les projeté dans un cas simple dont on a une référence.

## IX- Première estimation d'erreur

Dans un premier temps, nous allons nous intéresser à la précision atteinte par nos IMUs. Pour rappel, nous disposons actuellement des acquisitions de 2 capteurs : le MPU6050 et la NavChip ISNC02. Notre première plateforme expérimentale est le bras rotatif présenté en *Seq.VIII*.

### 1- Des sorties de capteurs aux mesures

La première étape est de convertir les sorties des 2 capteurs en données acquises en mesures  $\mathbf{y}_a$  pour l'accéléromètre et  $\mathbf{y}_\omega$  pour le gyromètre selon le cheminement présenté sur la *Fig.17*.

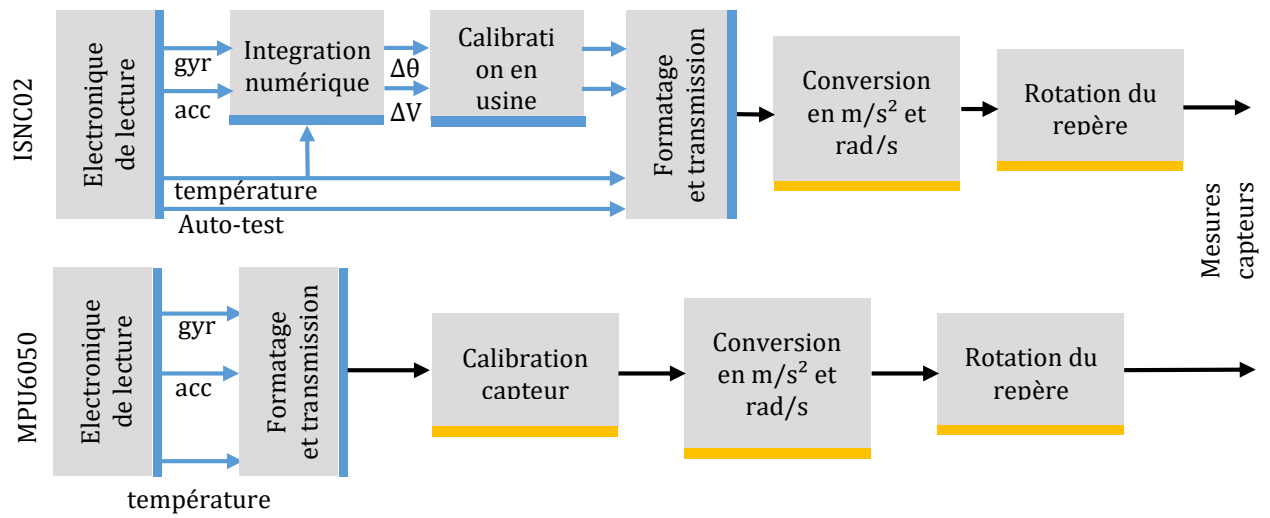


Fig.17 : Cheminement de l'obtention des mesures des capteurs pour le MPU6050 et le ISNC02

Nous supposons dans un premier temps que les 2 capteurs sont calibrés. Nous discuterons la question de calibration par la suite *Seq.X*.

Les données acquises sont, ensuite, analysées à l'aide de la fonction DR (*Seq. IV-2*) et la fonction EKF (*Seq.IV-3*). Nous comparons, ensuite, les résultats obtenus aux trajectoires et orientation obtenues via notre simulateur de table tournante *Seq.VIII-6*.

### 2- Réglage des filtres de Kalman

Dans cette partie, nous allons déterminer les matrices de covariance de mesures  $\mathbf{R}_a$  et  $\mathbf{R}_\omega$  et les matrices de covariance d'état  $\mathbf{Q}_a$  et  $\mathbf{Q}_\omega$ .

#### a- Matrice de covariance de mesures $\mathbf{R}_a$ et $\mathbf{R}_\omega$

Nous faisons des acquisitions statiques des accéléromètres et des gyromètres du MPU6050 et de l'ISNC02. Nous obtenons les histogrammes de mesures *Fig.18* et *Fig.19* respectivement.

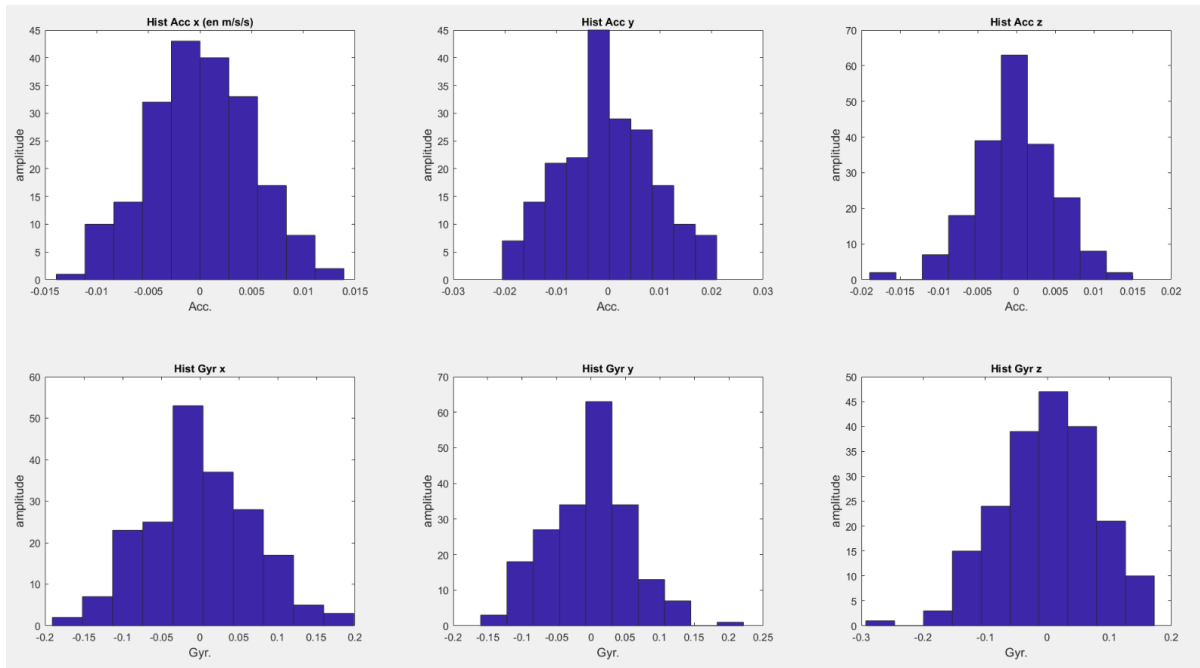


Fig.18 : Histogramme du bruit de l'accéléromètre et du gyromètre du MPU6050

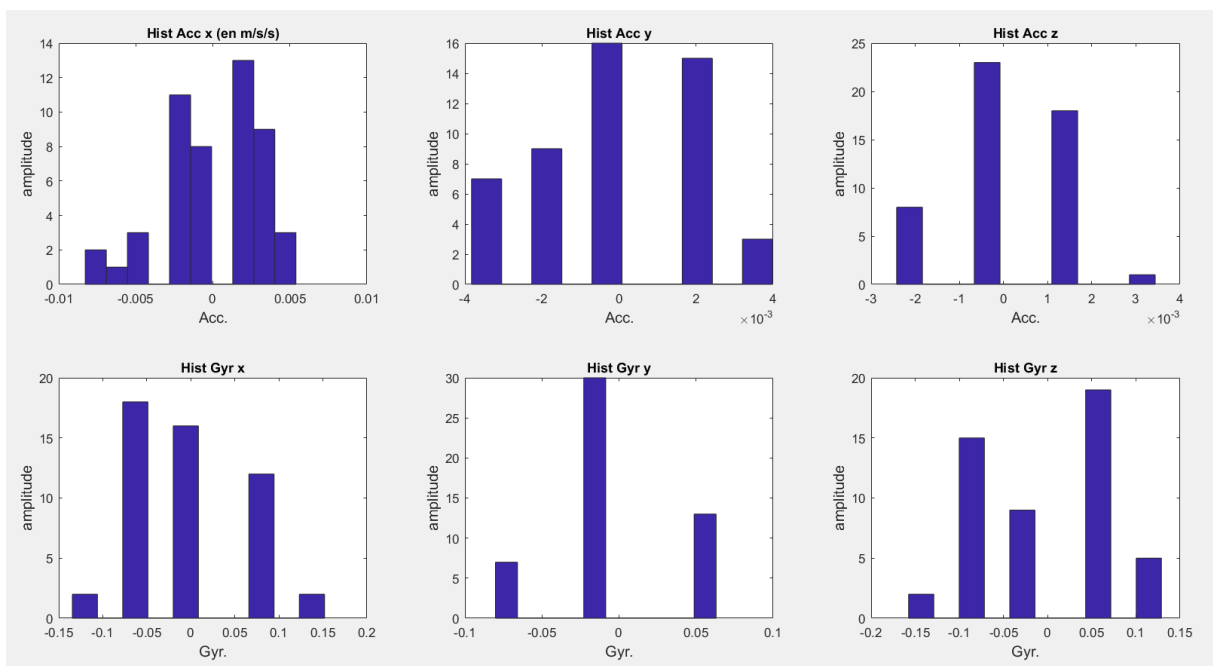


Fig.19 : Histogramme du bruit de l'accéléromètre et du gyromètre du ISNC02

Comme on peut le remarquer sur la Fig.18, les bruits de mesures sont bien gaussiens pour le MPU6050. Cette constatation est moins évidente pour le ISNC02 Fig.19. En effet les plages de mesures préréglées de  $\pm 16g$  et  $\pm 2000^\circ/s$  sont inadaptées aux valeurs faibles de bruit. Nous nous retrouvons, donc, avec des paliers bien distincts de valeurs pour les bruits de mesure (analogue à un signal fortement sous échantillonné). Néanmoins, les bruits de mesures du ISNC02 peuvent être assimilé à un bruit gaussien centré.

Nous mesurons les écarts types des 2 bruits et obtenons :

	NavChip	MPU6050
<b>Accéléromètre (en m/s<sup>2</sup>)</b>	[0.0091 ; 0.0062 ; 0.011]	[0.029 ; 0.035 ; 0.041]
<b>Gyromètre (en rad/s)</b>	[0.00080 ; 0.00094 ; 0.00092]	[0.00080 ; 0.00092 ; 0.00096]

Nous écrivons alors,

	NavChip	MPU6050
<b><math>R_a</math></b>	$\begin{bmatrix} 0.0091^2 & 0 & 0 \\ 0 & 0.0062^2 & 0 \\ 0 & 0 & 0.011^2 \end{bmatrix}$	$\begin{bmatrix} 0.029^2 & 0 & 0 \\ 0 & 0.035^2 & 0 \\ 0 & 0 & 0.041^2 \end{bmatrix}$
<b><math>R_\omega</math></b>	$\begin{bmatrix} 0.00080^2 & 0 & 0 \\ 0 & 0.00094^2 & 0 \\ 0 & 0 & 0.00092^2 \end{bmatrix}$	$\begin{bmatrix} 0.00080^2 & 0 & 0 \\ 0 & 0.00092^2 & 0 \\ 0 & 0 & 0.00096^2 \end{bmatrix}$

### b- Matrice de covariance d'état $Q_a$ et $Q_\omega$

On suppose que :

*erreur de modèle*

= *erreur discrétisation*

+ *éloignement de la trajectoire du bras rotatif d'une trajectoire circulaire*

Comme précisé dans *Seq.IV-3-f*, L'erreur de discrétisation est de l'ordre de 0.01mm pour la position et 0.001° pour l'orientation. Nous avons vu dans la partie *Seq.VIII-3*, que le bras rotatif tourne autour d'un axe  $\vec{u}$  fixe, incliné d'angles (0.01569°, 20°, 0.6468°) par rapport à la verticale.

Nous pouvons calculer alors que

$$Q_\omega = \begin{bmatrix} 9.1 \times 10^{-5} + 1.7444 \times 10^{-5} & 0 & 0 \\ 0 & 1.7444 \times 10^{-5} & 0 \\ 0 & 0 & 0.0037 + 1.7444 \times 10^{-5} \end{bmatrix}$$

$$Q_\omega = \begin{bmatrix} 1.08 \times 10^{-4} & 0 & 0 \\ 0 & 1.7444 \times 10^{-5} & 0 \\ 0 & 0 & 0.0037 \end{bmatrix}$$

$$Q_a = \begin{bmatrix} 0.000648 & 0 & 0 \\ 0 & 1.046 \times 10^{-5} & 0 \\ 0 & 0 & 0.0022 \end{bmatrix}$$

A noter que ces der matrices restent un paramètre de réglage du Kalman à rectifier afin d'améliorer les estimations. Les valeurs indiquées sont données à titre d'exemple.



### 3- Erreur d'estimation de pose

#### a- Erreurs d'estimation de pose pour une rotation continue

Nous programmons une rotation continue de la table tournante d'un angle  $\theta = 120^\circ$  avec une vitesse angulaire constante de  $10^\circ/s$ . Les fréquences d'échantillonnages des capteurs sont de 200Hz pour le MPU6050 et de 166Hz pour le ISNC02. Nous lançons 5 acquisitions similaires en moins de 15min. Nous supposons que la température est constante durant chaque acquisition et entre les différentes acquisitions. L'erreur moyenne maximale absolue (AME), et les écarts types d'erreurs (SD) entre les poses estimées et les poses simulées sont présentés dans le Tab.5 pour les orientations et le Tab.6 pour les positions.

Filtre	${}^d\varphi[^\circ]$		${}^d\theta[^\circ]$		${}^d\psi[^\circ]$	
	AME	SD	AME	SD	AME	SD
ISNC02 et EKF	0.16	0.0018	0.12	0.0018	0.20	0.0016
ISNC02 with DR	0.22	0.0020	0.07	0.0019	0.19	0.0027
MPU6050 with EKF	0.27	0.0029	0.13	0.0021	0.25	0.0032
MPU6050 with DR	0.29	0.0032	0.17	0.0022	0.27	0.0035

Tab.5 : Erreurs sur les estimations d'orientation pour des acquisitions de  $120^\circ$  du bras tournant.

Pour l'estimation d'orientation, les différents filtres/capteurs donnent de bons résultats avec une  $AME < 0.3^\circ$  pour une course continue de  $120^\circ$ .

Filtre	${}^d p_x [mm]$		${}^d p_y [mm]$		${}^d p_z [mm]$	
	AME	SD	AME	SD	AME	SD
ISNC02 et EKF	2.9	0.032	1.7	0.025	2.9	0.036
ISNC02 with DR	3.0	0.039	1.6	0.039	3.2	0.041
MPU6050 with EKF	12	0.097	9.1	0.080	14	0.093
MPU6050 with DR	96	0.093	61	0.084	102	0.10

Tab.6 : Erreurs sur les estimations de position pour des acquisitions de  $120^\circ$  du bras tournant.

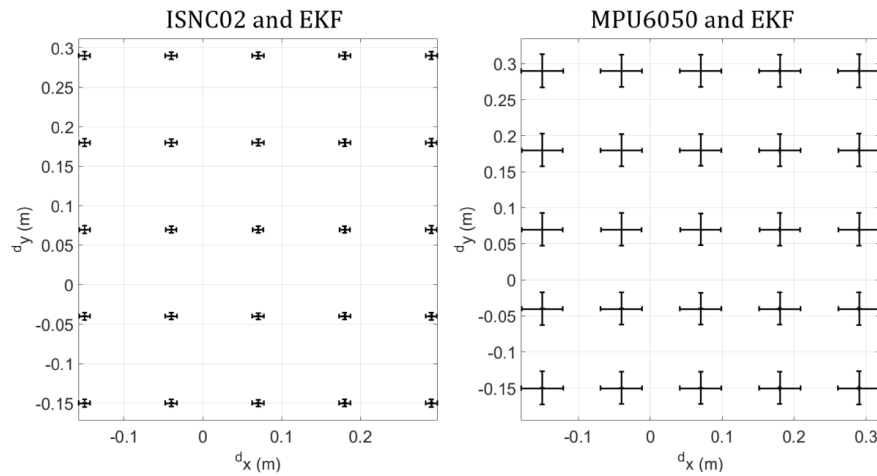
Pour la position, le MPU6050 ( $AME \approx 1cm$  pour l'EKF et  $AME \approx 1dm$  pour le DR) donne des estimations médiocres en comparaison avec l'ISNC02 ( $AME \leq 3.2mm$  pour les 2 algorithmes).

La prise en compte d'un modèle dans le cadre de l'EKF permet cependant de réduire l'erreur de poursuite du MPU6050 de 80%.

La raison que j'avance pour la mauvaise performance de l'accéléromètre du MPU6050 est sa mauvaise calibration. En effet, s'il est facile de débiaiser les mesures des gyromètres (acquisition des valeurs de biais quand les gyromètres sont statiques), il est plus difficile de caractériser les biais des accéléromètres vu que ce dernier lis le vecteur gravité s'il est statique.

Un biais de l'ordre  $0.001m/s^2$  par exemple, entraîne par application numérique une erreur de 0.1m pour la position pour une acquisition de 10s. Nous adresserons une solution de calibration dans la partie Seq.X.

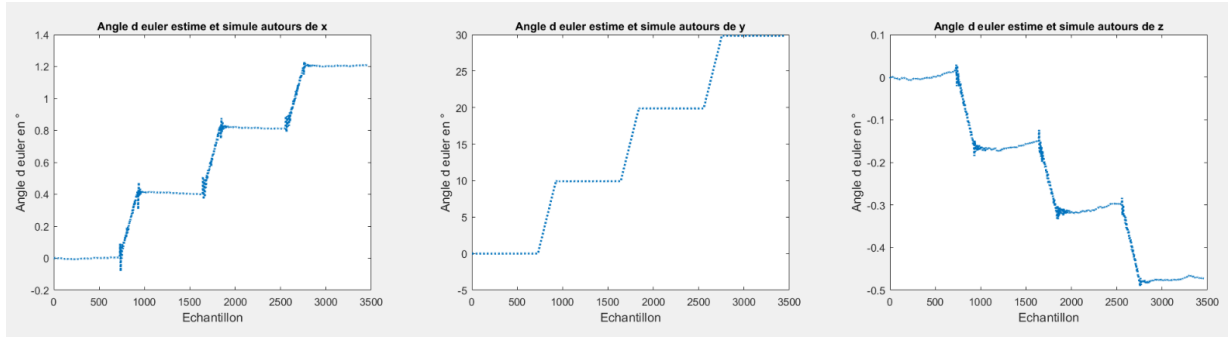
Si on propage les erreurs obtenues en sortie des EKF pour les 2 capteurs selon la formule *Seq.VII-1*, et le scénario *Seq.VII-4*, nous obtenons une erreur maximale absolue totale sur les projections 2D de **(7.4mm ; 4.7mm)** pour l'ISNC02 et de **(29mm; 23mm)** pour le MPU6050. Les barres d'erreurs sont affichées sur la *Fig.19*.



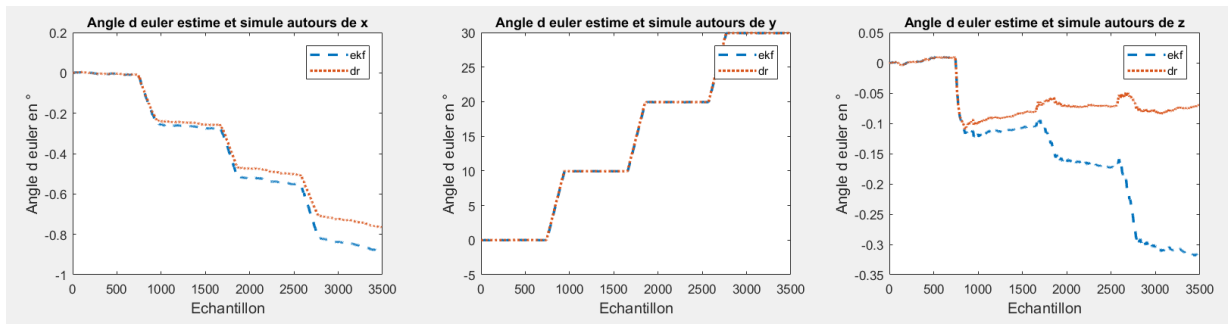
*Fig. 19 : Impact des erreurs d'estimation pour une rotation continue de 120° sur les projetés 2D*

### **b- Remarque sur l'erreur d'estimation pour une rotation discontinue**

Les fréquences d'échantillonnage de l'ISNC02 et du MPU6050 sont de 200Hz. Nous obtenons les angles suivants exprimés dans le repère du solide lié aux 2 capteurs. Les angles du NavChip ont été estimés par DR et ceux du MPU6050 par DR et EKF.



*Fig.19 : Angles estimés par DR à partir des mesures du NavChip*



*Fig.20 : Angles estimés par DR et par EKF à partir des mesures du MPU6050*

Qualitativement, le NavChip marque bien les arrêts et les redémarrages sur le 3 angles. Nous retrouvons 4 paliers réguliers. L'estimation MPU6050-DR transcrit très mal les paliers en z (composante tangentielle à la rotation principale et la plus bruitée). Ce résultat est amélioré par l'EKF.

# X- Calibration :

## 1- Equations de calibrations

Dans cette partie, je vais présenter un méthode de calibration sans équipement particulier basée sur [ref\(http://tesi.cab.unipd.it/44087/1/Thesis.pdf\)](http://tesi.cab.unipd.it/44087/1/Thesis.pdf).

**Remarque : Certaines notations sont redéfinies dans cette partie car elle a été rédigée comme une partie indépendante, reprise du code de la stagiaire de Hachème.**

Pour rappel, la calibration des capteurs inertiels consiste à trouver le gain, le biais et les coefficients d'influence entre axes des capteurs. On admet les modèles suivants pour les capteurs :

- **Mesure de l'accéléromètre  $y_a$**

$$y_a = M_a C_a a + \delta_a^b + n_a^b$$

- $M_a$  : matrice de gain de l'accéléromètre
- $C_a$  : matrice d'influence entre axes de l'accéléromètre
- $\delta_a^b$  : biais constant de l'accéléromètre
- $n_a^b$  : bruit blanc gaussien centré de l'accéléromètre
- $a$  : Accélération lue par un capteur idéalement

$$a = C_a^{-1} M_a^{-1} (y_a - \delta_a^b - n_a^b)$$

On pose  $T^a = C_a^{-1}$ ,  $K^a = M_a^{-1}$ ,  $b^a = -\delta_a^b$  et  $n^a = -n_a^b$

$$\boxed{a = T^a K^a (y_a + b^a + n^a)}$$

$$T^a = \begin{bmatrix} 1 & -\alpha_{yz} & \alpha_{zy} \\ \alpha_{zx} & 1 & -\alpha_{zx} \\ -\alpha_{zy} & \alpha_{yz} & 1 \end{bmatrix} : \text{Matrice des influences entre axes (cross-axis) à déterminer}$$

$$K^a = \begin{bmatrix} s_x^a & 0 & 0 \\ 0 & s_y^a & 0 \\ 0 & 0 & s_z^a \end{bmatrix} : \text{Matrice de gain à déterminer}$$

$$b^a = \begin{bmatrix} b_x^a \\ b_y^a \\ b_z^a \end{bmatrix} : \text{Biais}$$

$n^a$  : Bruit de mesure de l'accéléromètre bruit blanc gaussien centré i.i.d

On écrit :

$$\theta^a = [\alpha_{yz}, \alpha_{zy}, \alpha_{zx}, s_x^a, s_y^a, s_z^a, b_x^a, b_y^a, b_z^a]$$

et :

$$\boxed{a = h(y_a, \theta^a) = T^a K^a (y_a + b^a)}$$

Remarque : nous ne retrouvons pas le terme de bruit dans la dernière équation car nous allons manipuler des mesures d'accéléromètre moyennées sur des intervalles statiques.

- **Mesure du gyromètre  $y_\omega$  :**

On a

$$y_\omega = M_\omega C_\omega \omega + \delta_\omega^b + n_\omega^b$$

- $M_\omega$  : matrice de gain du gyromètre
- $C_\omega$  : matrice d'influence entre axes du gyromètre
- $\delta_\omega^b$  : Biais constant du gyromètre
- $n_\omega^b$  : bruit blanc gaussien centré du gyromètre
- $\omega$  : Vitesse angulaire lue par un capteur idéalement

$$\omega = C_\omega^{-1} M_\omega^{-1} (y_\omega - \delta_\omega^b - n_\omega^b)$$

On pose  $T^g = C_\omega^{-1}$ ,  $K^g = M_\omega^{-1}$ ,  $b^g = -\delta_\omega^b$  et  $n^g = -n_\omega^b$

On écrit :

$$\omega = T^g K^g (y_\omega + b^g + n^g)$$

$$T^g = \begin{bmatrix} 1 & -\gamma_{yz} & \gamma_{zy} \\ \gamma_{zx} & 1 & -\gamma_{zx} \\ -\gamma_{zy} & \gamma_{yz} & 1 \end{bmatrix} : \text{matrice des influences entre axes (cross-axis)}$$

$$K^g = \begin{bmatrix} s_x^g & 0 & 0 \\ 0 & s_y^g & 0 \\ 0 & 0 & s_z^g \end{bmatrix} : \text{matrice de gain}$$

$$b^g = \begin{bmatrix} b_x^g \\ b_y^g \\ b_z^g \end{bmatrix} : \text{biais}$$

$n^g$  : bruit de mesure du gyromètre bruit blanc gaussien centré i.i.d

On écrit :

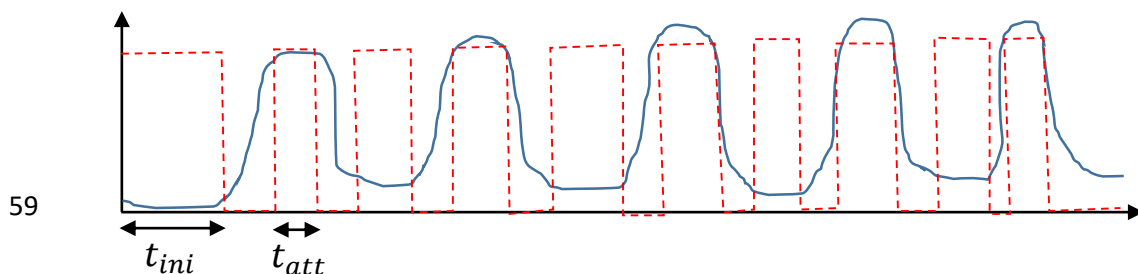
$$\Theta^g = [\gamma_{yz}, \gamma_{zy}, \gamma_{zx}, s_x^g, s_y^g, s_z^g, b_x^g, b_y^g, b_z^g]$$

## 2- Calibration sans équipement

Dans cette partie nous allons proposer un algorithme de calibration de capteurs qui ne nécessite pas d'équipement particulier ainsi qu'un aperçu des résultats obtenus par ce dernier.

### Signal de la calibration :

Cette méthode de calibration est basée sur un enchainement d'intervalles de mouvement et d'intervalles statiques. Sont requises entre 36 et 50 orientations différentes des capteurs (intervalles statiques différents) avec des temps d'attente  $1s \leq t_{wait} \leq 4s$ . Cela permet d'obtenir un bon compromis entre la précision de calibration, la stabilité du biais du gyromètre et la reduction de bruit. Cette valeur a été déterminée empiriquement. Le [signal de calibration](#) se présente schématiquement comme sur la Fig.21.



### Comment détermine-t-on les intervalles statiques ?

On pose :  $\zeta(t) = \sqrt{[var_{t\omega}(a_x^t)]^2 + [var_{t\omega}(a_y^t)]^2 + [var_{t\omega}(a_z^t)]^2}$

avec :  $var_{t\omega}(a^t)$  : la variance de  $a^t$  sur un intervalle de durée  $t\omega$  centrée en  $t$

Un échantillon à  $t$  est considéré comme « statique » si  $\zeta(t) < seuil$

Une séquence ininterrompue de  $t_{est}$  considérée comme un intervalle statique.

On définit le seuil comme un multiple de la variance du signal de l'accéléromètre à  $t_{ini}$

#### - Processus de calibration :

1. On lance l'acquisition et on attend  $t_{init} = 40s$ .

```
fe = 100; % fréquence d'échantillonnage (en Hz)
Tinit = 40; % temps d'initialisation (en s)
TintervTinit = fe*Tinit; % Nombre d'échantillons d'initialisation
```

2. En moyennant les mesures du gyromètres dans cet intervalle nous obtenons des **estimés des biais du gyromètre** :  $b_x^g, b_y^g, b_z^g$ . (dans le code BiaiseGx, BiaiseGy, BiaiseGz respectivement)

```
% Récupération des données
GX = Data.Gx; % mesure selon x
GY = Data.Gy; % mesure selon y
GZ = Data.Gz; % mesure selon x
% Conversion des sorties du gyromètre en unité SI (rad/s)
Remarque il faudrait mettre le facteur de conversion en paramètre
d'entrée.
ValGx = double(GX*250)/32000;
ValGy = double(GY*250)/32000;
ValGz = double(GZ*250)/32000 ;
% Calcul du biais comme moyenne des signaux sur un intervalle
statique :
```

```
BiaiseGx = mean(ValGx(1:TintervTinit), 'double');
BiaiseGy = mean(ValGy(1:TintervTinit), 'double');
BiaiseGz = mean(ValGz(1:TintervTinit), 'double');
% Retirer le biais des sorties du gyromètre
BFGx = ValGx-BiaiseGx;
BFGy = ValGy-BiaiseGy;
BFGz = ValGz-BiaiseGz;
```

Il nous reste à estimer :

$$\theta^g = [\gamma_{yz}, \gamma_{zy}, \gamma_{zx}, s_x^g, s_y^g, s_z^g]$$

En moyennant les mesures du gyromètres dans cet intervalle nous obtenons des **estimés des biais du gyromètre** :  $b_x^g, b_y^g, b_z^g$ . (dans le code BiaiseGx, BiaiseGy, BiaiseGz respectivement)

3. Détermination de la variance des données de l'accéléromètre dans  $T_{init}$  (dans l'intervalle initial). On va récupérer les données de l'accéléromètre puis choisir de calculer une variance glissante sur une fenêtre de taille  $N$  dans l'intervalle initial.

```

% On choisir un largeur pour notre fenêtre glissante
Remarque : Il faut que N soit inférieur un temps d'attente
N = TintervTinit/50;
% Récupération des données de l'accéléromètre :
Remarque : Faire en sorte que le tableau de données soit au bon
format (définir bon format)
AX = Data.Ax;
AY = Data.Ay;
AZ = Data.Az;
% Conversion des sorties de l'accéléromètre en unité SI (g)
Remarque il faudrait mettre le facteur de conversion en paramètre
d'entrée.
ValAx = double(AX*2)/32000;
ValAy = double(AY*2)/32000;
ValAz = double(AZ*2)/32000;
% Calcul moyenne glissante (afin de calculer la variance)
Remarque : formule :

$$\bar{x}_n = \frac{1}{N} \sum_{k=0}^{N-1} x_{n-k}, \text{ ou bien } \bar{x}_n = \bar{x}_{n-1} + \frac{x_n - x_{n-N}}{N}$$

Plus d'infos sur la moyenne glissante :
https://fr.wikipedia.org/wiki/Moyenne\_mobile
MAxGliss = conv(ValAx, (1/N)*ones(1,N), 'same');
MAyGliss = conv(ValAy, (1/N)*ones(1,N), 'same');
MAzGliss = conv(ValAz, (1/N)*ones(1,N), 'same');
% Calcul de la variance glissante
VarAxGliss = conv((ValAx-MAxGliss).^2, (1/N)*ones(1,N), 'same');
VarAyGliss = conv((ValAy-MAyGliss).^2, (1/N)*ones(1,N), 'same');
VarAzGliss = conv((ValAz-MAzGliss).^2, (1/N)*ones(1,N), 'same');
% Calcul de la variance sur Twait :
formule  $var = \sqrt{var(a_x^t)^2 + var(a_y^t)^2 + var(a_z^t)^2}$ 
ValVar = sqrt(VarAxGliss.^2+VarAyGliss.^2+VarAzGliss.^2);
% Calcul de la variance du T_init
VarInit = mean(ValVar(N+1:TintervTinit-N));

```

A la fin de cette étape on obtient la variance de l'accélération dans l'intervalle  $T_{init}$  :  $s_{init}$  (dans le code VarInit)

Pour déterminer si un intervalle est statique ou non, nous calculons la valeur

$$s(t) = \sqrt{var_{t_\varepsilon}(a_x^t)^2 + var_{t_\varepsilon}(a_y^t)^2 + var_{t_\varepsilon}(a_z^t)^2}$$

Avec  $var_{t_\varepsilon}(a^t)$  la variance de  $a^t$  dans un intervalle de durée  $t_\varepsilon$  centré en  $t$

Un échantillon à  $t$  est considéré statique si la variance  $s(t) < seuil = k \times s_{init}$  (dans le code  $k = 1.7$ ). Un intervalle statique est une séquence ininterrompue de  $t$ .

$y$  est l'indice qui détermine si un intervalle est dynamique ou statique.

```
% calcul du seuil
```

```

seuil = 1.7*VarInit;

% Grandeur de la variance <= le seuil détecteur = 1
s_interval = (ValVar<=seuil);

% Détermination d'indices qui indiquent le changement d'état
(dynamique ou statique)
y = find(diff(s_interval));

```

4. **Calibration de l'accéléromètre** : On tourne le capteur dans  $M$  positions distinctes avec  $M \in [36, 50]$  tout en stabilisant le capteur à chaque rotation et on extrait  $M$  vecteurs de mesures d'accélérations moyennées sur chaque intervalle statique (ici de longueur 10) :  $y_{a,k}$

```

% La moyenne sur l'intervalle statique :
for k = 1:10
    MeanAx(k) = mean(ValAx(y(2*k-1):y(2*k)));
    MeanAy(k) = mean(ValAy(y(2*k-1):y(2*k)));
    MeanAz(k) = mean(ValAz(y(2*k-1):y(2*k)));
end

```

5. On sait qu'en statique l'accéléromètre mesure  $g$ . Donc, la norme de la somme mesures d'un accéléromètre parfaitement calibrée  $a$  selon les différents axes doit être égale à  $\|g\|$  norme du vecteur gravité.

Par conséquent, pour trouver  $\theta^a$ , on minimise la fonction  $L(\theta^a)$  :

$$L(\theta^a) = \sum_{k=1}^M (\|g\|^2 - \|h(y_a, \theta^a)\|^2)^2$$

avec :

- $h$  la fonction qui corrige les mesures en appliquant les paramètres préestimé,
- $y_a$  : les valeurs lues pas les accéléromètres.
- $M$  : les nombre d'intervalles statiques.
- Pour cela nous utilisons l'algorithme de LM Levenbeg-Marquat.

```

% Gravité de la terre locale
% glocal = 9.80573 ;
% lg = 9,80665 ;
% glocal = 9.80573/9.80665;      % en g

% Choix de l'algorithme de Levenberg Marquardt pour la minimisation
options.Algorithm = 'levenberg-marquardt';

% x(1:9):AlphYZ,AlphZY,AlphZX,SxAcc,SyAcc,SzAcc,BxAcc,ByAcc,BzAcc
funAcc = @(ParaAcc) (1.^2*ones(9,1) - ((ParaAcc(4) * (MeanAx(1:9) .'+ParaAcc(7)) -
ParaAcc(1) * ParaAcc(5) * (MeanAy(1:9) .'+ParaAcc(8)) + ParaAcc(2) * ParaAcc(6) * (MeanAz(1:9) .'+ParaAcc(9))) .^2 + (ParaAcc(5) * (MeanAy(1:9) .'+ParaAcc(8)) -
ParaAcc(3) * ParaAcc(6) * (MeanAz(1:9) .'+ParaAcc(9))) .^2 + (ParaAcc(6) * (MeanAz(1:9) .'+ParaAcc(9))) .^2)) .^2);

% Initialisation : on suppose que les 3 axes sont orthogonaux et qu'il n'y
a pas d'erreur d'échelle, ni de biais

% Minimisation (https://fr.mathworks.com/help/optim/ug/lsgnnonlin.html)

```

```

[ParaAcc, resnorm, residual, exitflag, output] =
lsqnonlin(funAcc, [0,0,0,1,1,1,0,0,0].', [], [], options);

% Récupération des données
Ta = [1 -ParaAcc(1) ParaAcc(2); 0 1 -ParaAcc(3); 0 0 1];
Ka = [ParaAcc(4) 0 0 ; 0 ParaAcc(5) 0; 0 0 ParaAcc(6)];
Ba = [ParaAcc(7) ParaAcc(8) ParaAcc(9)].';

ValAcc = [ValAx.'; ValAy.'; ValAz.'];

AO = Ta*Ka*(ValAcc+Ba);

% Moyenne sur les différents intervalles (?)
for k = 1:10
    MeanAxO(k) = mean(AO(1, Y(2*k-1):Y(2*k)));
    MeanAyO(k) = mean(AO(2, Y(2*k-1):Y(2*k)));
    MeanAzO(k) = mean(AO(3, Y(2*k-1):Y(2*k)));
end

```

A la fin de cette étape on obtient un estimé de :

$$\Theta^a = [\alpha_{yz}, \alpha_{zy}, \alpha_{zx}, s_x^a, s_y^a, s_z^a, b_x^a, b_y^a, b_z^a]$$

6. **Calibration du gyromètre :** On définit une fonction  $\xi$  qui prend pour entrée N mesures de gyromètre  $\mathbf{y}_\omega$  et  $u_{a,0}$  le quaternion unitaire de  $\mathbf{g}$  renvoyé après correction des données de l'accéléromètre à l'instant initial.

$$u_{g,k} = \xi[\boldsymbol{\omega}, u_{a,0}]$$

- $u_{g,k}$  est le quaternion unitaire de  $\mathbf{g}$  renvoyé par le gyromètre à l'instant k.
- $\xi$  Peut être n'importe que algorithmes qui intègre les vitesses angulaires des phases dynamiques en orientations.

**Remarque :** Dans le code de la stagiaire il y a une erreur dans la construction du quaternion unitaire.

Pour construire  $\xi$ , on échantillonne l'équation de dérivée continue d'un quaternion :

$$f(q, t) = \dot{q} = \frac{1}{2} \boldsymbol{\Omega}(\boldsymbol{\omega}(t))q$$

avec

$$\boldsymbol{\Omega}(\boldsymbol{\omega}(t)) = \begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ -\omega_z & \omega_y & -\omega_x & 0 \end{bmatrix}$$

On échantillonne grâce à du RK4 (Runge-Kutta d'ordre 4) Pour ne pas perdre de précision à ce niveau :

$$a) \quad q_{k+1} = q_k + \Delta t \frac{1}{2} (k_1 + 2k_2 + 2k_3 + k_4)$$



b)  $k_i = f(q^{(i)}, t_k + c_i \Delta t)$   
 c)  $q^{(i)} = q_k$ , pour  $i = 1$  et  $q^{(i)} = q_k + \Delta t \sum_{j=1}^{i-1} a_{ij} k_j$ , pour  $i > 1$   
 avec  $c_1 = 0, c_2 = \frac{1}{2}, c_3 = \frac{1}{2}$  et  $c_4 = 1$   
 et  $a_{21} = \frac{1}{2}, a_{31} = 0, a_{41} = 0$  et  $a_{32} = \frac{1}{2}, a_{42} = 0, a_{43} = 1$

On doit aussi normaliser le quaternion à chaque étape :

$$q_{k+1} \rightarrow \frac{q_{k+1}}{\|q_{k+1}\|}$$

**Remarque : Je n'ai pas encore relu le code implémenté par la stagiaire pour la RK**

Puis on multiplie ce quaternion par le quaternion de gravité initial lu par l'accéléromètre calibré.

7. Les mesures d'un gyromètre parfaitement calibré doivent conduire à un quaternion estimé du vecteur gravité parfaitement égal à celui estimé par un accéléromètre idéal. Par conséquent, pour trouver  $\theta^g$ , on minimise la fonction  $L(\theta^g)$  :

$$L(\theta^g) = \sum_{k=2}^M \|u_{a,k} - u_{g,k}\|^2$$

$$L(\theta^g) = \sum_{k=2}^M \|u_{a,k} - \xi[\omega, u_{a,k-1}]\|^2$$

A la fin de cette étape on obtient un estimé de :

$$\theta^g = [\gamma_{yz}, \gamma_{zy}, \gamma_{zx}, s_x^g, s_y^g, s_z^g, b_x^g, b_y^g, b_z^g]$$

### 3- Test de l'algorithme

Pour tester cet algorithme de calibration, mon idée serait d'utiliser un capteur comme le Yost-lab (**partie III-3 et Annexe 1**) qui nous fournit et les valeurs calibrées en usine et les valeurs non calibrés. On opère notre calibration manuelle puis on compare les valeurs obtenues aux valeurs d'usine.

Nous pouvons, pour l'instant, montrer les résultats de la publication [ref] qui semblent assez prometteuses :

Calibration d'usine	Calibration algorithme
---------------------	------------------------

(a) Scaling - Accelerometer	(b) Scaling - Gyroscope	(a) Estimated Scaling - Acc	(b) Estimated Scaling - Gyro																																				
<table border="1"><tr><td>415</td><td>0.00</td><td>0.00</td></tr><tr><td>0.00</td><td>413</td><td>0.00</td></tr><tr><td>0.00</td><td>0.00</td><td>415</td></tr></table>	415	0.00	0.00	0.00	413	0.00	0.00	0.00	415	<table border="1"><tr><td>4778</td><td>0.00</td><td>0.00</td></tr><tr><td>0.00</td><td>4758</td><td>0.00</td></tr><tr><td>0.00</td><td>0.00</td><td>4766</td></tr></table>	4778	0.00	0.00	0.00	4758	0.00	0.00	0.00	4766	<table border="1"><tr><td>414.41</td><td>0</td><td>0</td></tr><tr><td>0</td><td>412.05</td><td>0</td></tr><tr><td>0</td><td>0</td><td>414.61</td></tr></table>	414.41	0	0	0	412.05	0	0	0	414.61	<table border="1"><tr><td>4778.0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>4764.8</td><td>0</td></tr><tr><td>0</td><td>0</td><td>4772.6</td></tr></table>	4778.0	0	0	0	4764.8	0	0	0	4772.6
415	0.00	0.00																																					
0.00	413	0.00																																					
0.00	0.00	415																																					
4778	0.00	0.00																																					
0.00	4758	0.00																																					
0.00	0.00	4766																																					
414.41	0	0																																					
0	412.05	0																																					
0	0	414.61																																					
4778.0	0	0																																					
0	4764.8	0																																					
0	0	4772.6																																					
(c) Misalignment - Accelerometer	(d) Misalignment - Gyroscope	(c) Estimated Misalignment - Acc	(d) Estimated Misalignment - Gyro																																				
<table border="1"><tr><td>1.00</td><td>0.00</td><td>-0.01</td></tr><tr><td>0.01</td><td>1.00</td><td>0.01</td></tr><tr><td>0.02</td><td>0.01</td><td>1.00</td></tr></table>	1.00	0.00	-0.01	0.01	1.00	0.01	0.02	0.01	1.00	<table border="1"><tr><td>1.00</td><td>-0.01</td><td>-0.02</td></tr><tr><td>0.00</td><td>1.00</td><td>0.04</td></tr><tr><td>-0.01</td><td>0.01</td><td>1.00</td></tr></table>	1.00	-0.01	-0.02	0.00	1.00	0.04	-0.01	0.01	1.00	<table border="1"><tr><td>1.0000</td><td>-0.0066</td><td>-0.0110</td></tr><tr><td>0.0102</td><td>1.0001</td><td>0.0114</td></tr><tr><td>0.0201</td><td>0.0098</td><td>0.9998</td></tr></table>	1.0000	-0.0066	-0.0110	0.0102	1.0001	0.0114	0.0201	0.0098	0.9998	<table border="1"><tr><td>0.9998</td><td>-0.0149</td><td>-0.0218</td></tr><tr><td>0.0003</td><td>1.0007</td><td>0.0433</td></tr><tr><td>-0.0048</td><td>0.0121</td><td>1.0004</td></tr></table>	0.9998	-0.0149	-0.0218	0.0003	1.0007	0.0433	-0.0048	0.0121	1.0004
1.00	0.00	-0.01																																					
0.01	1.00	0.01																																					
0.02	0.01	1.00																																					
1.00	-0.01	-0.02																																					
0.00	1.00	0.04																																					
-0.01	0.01	1.00																																					
1.0000	-0.0066	-0.0110																																					
0.0102	1.0001	0.0114																																					
0.0201	0.0098	0.9998																																					
0.9998	-0.0149	-0.0218																																					
0.0003	1.0007	0.0433																																					
-0.0048	0.0121	1.0004																																					

Table 4.8: MTi estimated calibration parameters.

Remarque : Pour la méthode de calibration classique, méthode du pendule se référer à : la partie 2. 1 Intrinsic calibration de <https://ap.isr.uc.pt/archive/19.pdf>

## XI- Initialisation de algorithmes (caméra et IMU)

# Annexe 1 : Fiche des capteurs utilisés

## 1- NavChip ISNC02

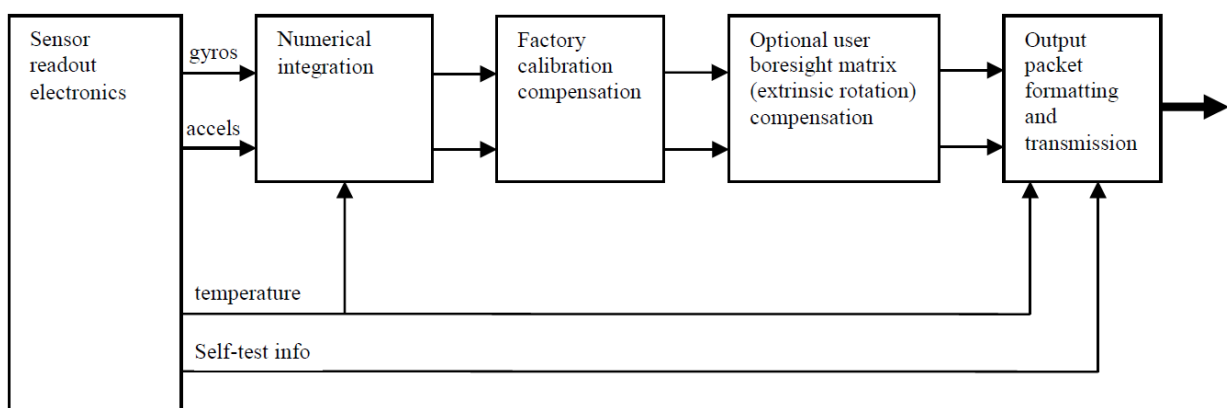
### Sorties

- $\Delta\theta$  rotation incrémentale sur la période d'intégration **analogue à une vitesse angulaire** mais une partie de l'intégration est effectuée pour filtrer les non linéarités  $\Delta\theta$  est aussi pertinente à 100Hz qu'à 1KHz.
  - Coning compensation si activée
  - Intégration haute précision les nombres tronqués sont stockés et ramenés au prochain échantillon
  - 0.00625mrad/bit ( 0.00036 °/bit) (+-32768 bits range donc max rotation par update est de 0.2048 rad par période)
- $\Delta V$  : intégrale de l'accélération sur une période de mesure **analogue à une accélération** mais une partie de l'intégration est effectuée pour filtrer les non linéarités  $\Delta V$  est aussi pertinente à 100Hz qu'à 1KHz.
  - Coning et Culing compensation si activées
  - Intégration haute précision les nombres tronqués sont stockés et ramenés au prochain échantillon
  - 39.0625\*10<sup>-6</sup> m/s ((+-32768 bits range donc max rotation 1.28m/s)

### Caractéristiques principales

- **Pré-calibré** : compensation des désalignements des axes : orthogonalité des axes garantie (mais pas orthogonalité à la plateforme, voir notes)
- Acquisition se fait à 1kHz mais Fréquence d'échantillonnage de sortie est de 100 Hz ou de **200Hz** (réglage à 100Hz si dynamique très rapide (20rad/s et 120m/s/s pour éviter le overflow à priori on reste à 200Hz)
- Stabilité du biais du gyromètre : 5°/h
- Angular random walk : 0.18°/√h
- Velocity random walk : 0.03m/s/√h

### Schéma bloc



### Notes importantes

- Les accélérations sont calculées dans un point situé à 1.5mm sous le centre de la NavChip (centre de l'empreinte sur la surface)
- mise à disposition d'un outil de calibration extrinsèque qui permet de calibrer le parallélisme de l'IMU et de la surface

## 2- Mti-1-8A7G6 (Xsens)

### Sorties :

- Mesures du gyromètre, de l'accéléromètre et du magnétomètre
- $\Delta q$  rotation incrémentale sur la période d'intégration **analogue à une vitesse angulaire** mais une partie de l'intégration est effectuée pour filtrer les non linéarités  $\Delta\theta$ .
  - Coning compensation si activée
- $\Delta V$  : intégrale de l'accélération sur une période de mesure **analogue à une accélération** mais une partie de l'intégration est effectuée pour filtrer les non linéarités  $\Delta V$ .
  - Coning et Culing compensation si activées

### Caractéristiques principales :

- **Pré-calibré** : Dans la ROM de chaque IMU sont stockés les valeurs du biais, l'alignement des axes avec PCB et entre eux, le gain en fonction de la température.
- Disponibilité des données d'un magnétomètre (permettrait une mesure supplémentaire d'orientation).
- L'origine des mesures se trouve à l'emplacement de l'accéléromètre
- Acquisition se fait à **800Hz** mais Fréquence d'échantillonnage de sortie est de **100 Hz max**
- Filtre passe bas de fréquence de coupure 255Hz
- Stabilité du biais du gyromètre :  $10^\circ/h$
- Densité de bruit gyromètre :  $0.007^\circ/s/\sqrt{Hz}$
- Densité de bruit accéléromètre:  $0.12g/\sqrt{Hz}$

### Schéma bloc

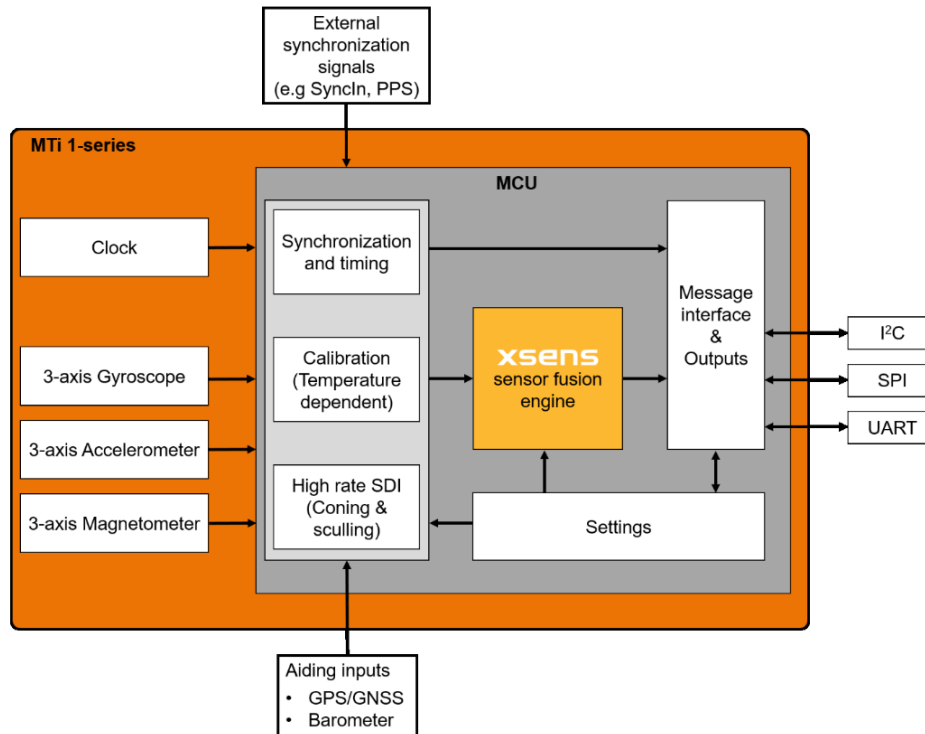


Figure 1: MTi 1-series module diagram

### Notes importantes

- Comprends un bloc fusion qui permet, à partir des données des différents capteurs (fusionnés entre

### 3 - Yost-Lab 3space

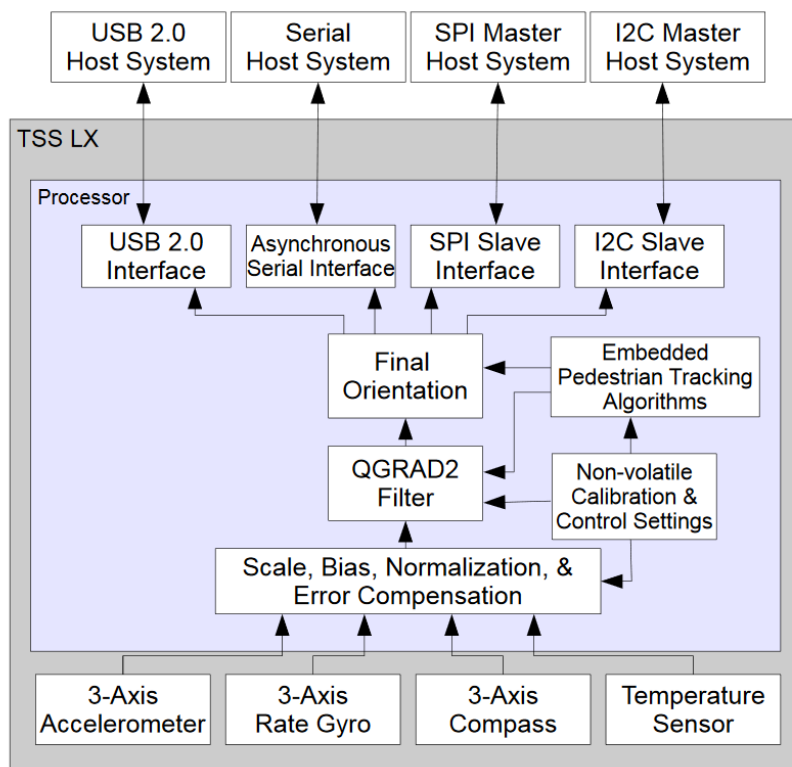
#### Sorties :

- Mesures brutes du gyromètre et de l'accéléromètre
- Mesures « normalisées » des capteurs (représentatifs des grandeurs physiques mesurées)
- Mesure de température
- Orientation : Peut être renvoyée en quaternions absolus ou relatifs, matrice de rotation ou angle d'Euler
- $\Delta q$  rotation incrémentale sur la période d'intégration **analogue à une vitesse angulaire** mais une partie de l'intégration est effectuée pour filtrer les non linéarités  $\Delta q$ . Permet de réduire le bruit
- $\Delta V$  : intégrale de l'accélération sur une période de mesure **analogue à une accélération** mais une partie de l'intégration est effectuée pour filtrer les non linéarités  $\Delta V$ . Permet de réduire le bruit

#### Caractéristiques principales

- Calibré en usine : Gain, biais, désalignement des axes pour le gyromètre et l'accéléromètre. (activable ou non)
- Filtre update rate jusqu'à 1750Hz
- Stabilité du biais du gyromètre :  $11^\circ/h$
- Densité de bruit gyromètre :  $0.007^\circ/s/\sqrt{Hz}$
- Densité de bruit accéléromètre: 0.000061g/LSB pour  $\pm 2g$       0.000122g/LSB pour  $\pm 4g$   
0.000244g/LSB pour  $\pm 8g$       0.000488g/LSB pour  $\pm 16g$

#### Schéma bloc



#### Notes importantes

- Personnalisable, filtre passe bas réglable, sur-échantillonnage, correction d'erreur
- Filtrage par filtre de Kalman étendu intégré (boîte noire).
- Peut envoyer des données à très haute fréquence

## 4 – MPU-6050

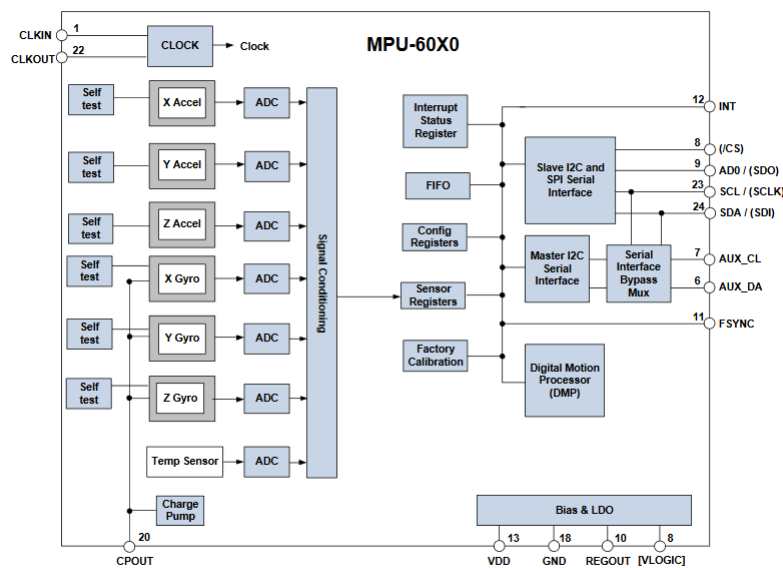
### Sorties

- Mesures du gyromètre, de l'accéléromètre
- Mesures de température
- Angles et positions renvoyée par un Digital Motion Processor intégré (DMP)

### Caractéristiques principales

- **Attention** pas de calibration usine donc le composant est sujet à des non linéarités, du désalignement d'axes. Il faut calibrer chaque composant sois même (pour plus d'informations sur les ordres de grandeur voir specs ou excel)
- Disponibilité d'un filtre passe bas configurable fréquence max de 256Hz pour le gyromètre et 260Hz pour l'accéléromètre
- Fréquences d'acquisition à 1KHz, 500Hz, 250Hz, ... 5Hz
- Stabilité du biais du gyromètre :  $11^\circ/h$
- Densité de bruit gyromètre :  $0.005^\circ/s/\sqrt{Hz}$  (Total RMS noise)
- Densité de bruit accéléromètre:  $0.4g/\sqrt{Hz}$  (@10Hz, AFS\_SEL=0 & ODR=1KHz)

### Schéma bloc



Note: Pin names in round brackets ( ) apply only to MPU-6000  
Pin names in square brackets [ ] apply only to MPU-6050

### Notes importantes

## 5- Angular random walk (ARW) et Velocity random walk

[ref http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.210.1133&rep=rep1&type=pdf](http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.210.1133&rep=rep1&type=pdf)

C'est une mesure de bruit homogène à du deg/sec (le bruit peut être défini par sa PSD (power spectral density) unité °/s/√Hz ou FFT (°/s)<sup>2</sup>/Hz.

**Définition :** Directement applicable au calcul d'angle découle du bruit et indépendante des gains et biais. Plus le temps d'intégration va augmenter plus ARW va augmenter

### Comment l'ARW affecte le calcul d'angles ?

Exemple : Un capteur à une ARW de 0.99 °/s

Après 1s : la déviation sera de 1°

Après 10s : la déviation sera de 9.9°

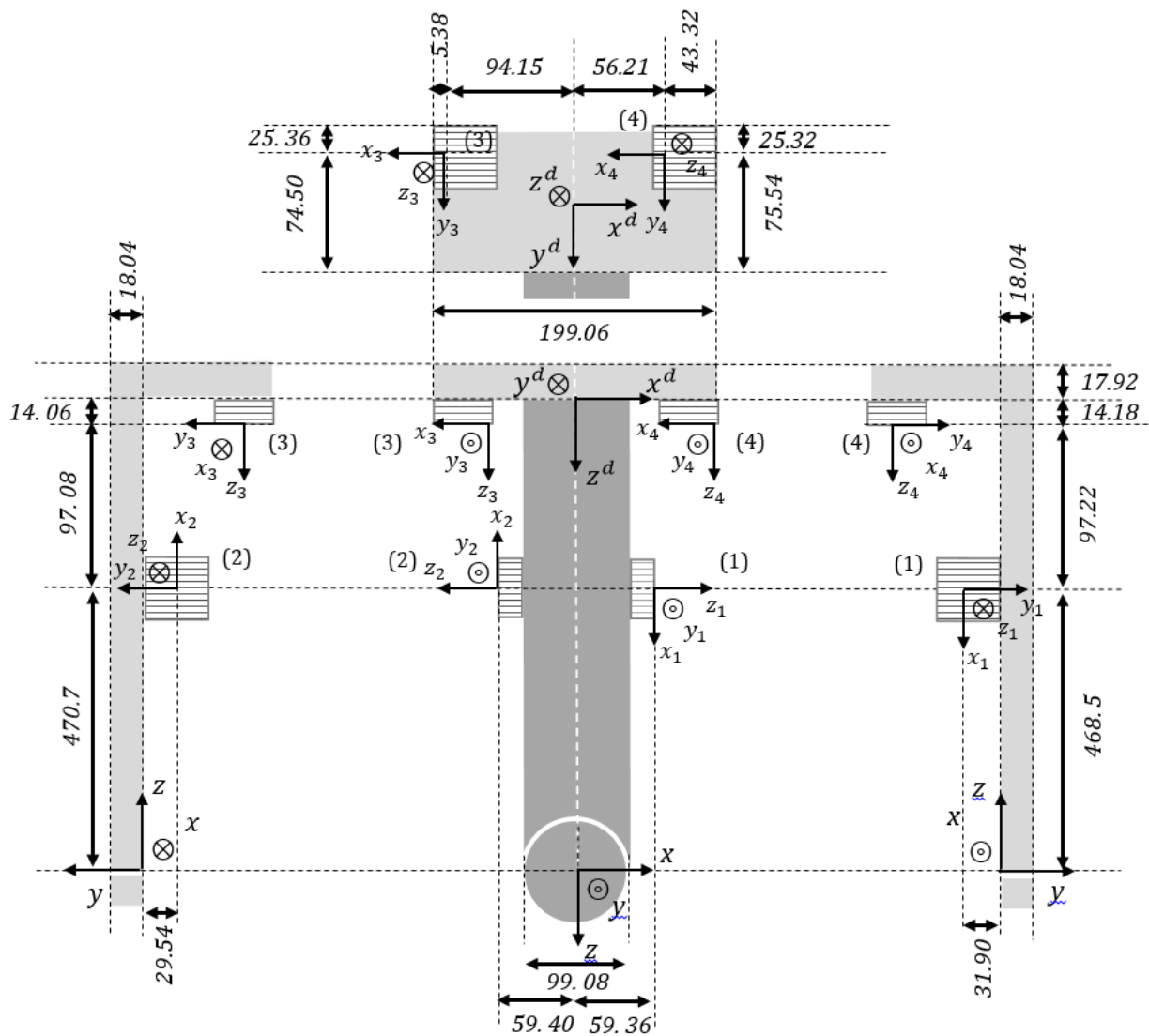
après 1000s : la déviation sera de 31.5°

### Conversion :

$$\text{PSD à ARW : } ARW(°/\sqrt{h}) = \frac{1}{60} \sqrt{PSD \left[ \left( \frac{°}{h} \right)^2 / Hz \right]}$$

$$\text{FFT à ARW : } ARW(°/\sqrt{h}) = \frac{1}{60} FFT \left[ \left( \frac{°}{h} \right) / \sqrt{Hz} \right]$$

## Annexe 2 : Alternatives au Filtre de Kalman étendu et comparaison



### Positions des modules dans le repère lié au bras (mm):

- (1) : (-59.36 ; 31.90 ; 468.5)
- (2) : (59.40 ; 29.54 ; 470.7)
- (3) : (94.15 ; 74.50 ; 567.78)
- (4) : (-56.21 ; 75.54 ; 565.72)

### Positions des modules dans le repère lié au détecteur (mm):

- (1) : (59.36 ; -56.14 ; 111.4)
- (2) : (-59.40 ; -58.50 ; 111.14)
- (3) : (-94.15 ; -25.07 ; 14.18)
- (4) : (56.21 ; -25.11 ; 14.06)

Poids : 3Kg

### Conversions repères

- (1) : (x:-z1 ; y:y1 ; z:-x1)
- (2) : (x:z2 ; y:y2 ; z:x2)
- (3) : (x:x3 ; y:y3 ; z:-z3)
- (4) : (x:x4 ; y:y4 ; z:-z4)



## Annexe 3 : Alternatives au Filtre de Kalman étendu et comparaison

Dans cet annexe nous allons brièvement présenter d'autres algorithmes d'estimation qui n'ont pas été retenus face au Kalman Étendu additif contraint pour l'estimation de la pose.

### Backward smoothing EKF (BSEKF)

**Idée :** la k-ième étape d'un problème de filtrage non linéaire peut être considérée comme une probabilité de maximum à posteriori (MAP) nous écrivons la fonction de densité de probabilité (PDF) comme :

$$p_k = \exp(-J_k)$$

$$J_k = \frac{1}{2} \sum_{i=0}^{k-1} \{ [\tilde{y}_{i+1} - h_{i+1}(x_{i+1})]^T R_{i+1}^{-1} [\tilde{y}_{i+1} - h_{i+1}(x_{i+1})] + w_i^T Q_i^{-1} w_i \} + \frac{1}{2} (x_0 - \hat{x}_0)^T P_0^{-1} (x_0 - \hat{x}_0)$$

$\hat{x}_k$  (estimé MAP): est le vecteur  $x_k$  qui, avec  $x_i$  et le bruit de processus  $w_i$  pour  $i = 0, 1, \dots, k-1$  minimise  $J_k$  soumis à l'équation dynamique

$$x_{i+1} = f_i(x_i, w_i) \text{ pour } i=0, 1, \dots, k-1$$

$Q_i$ : covariance du bruit du processus

$R_i$ : covariance du bruit de mesure

$\tilde{y}_{i+1}$ : mesure au temps  $t_{i+1}$

$h_{i+1}(x_{i+1})$  modèle de mesure non linéaire

$\hat{x}_0$  est l'estimation a posteriori de l'état avec une covariance  $P_0$

Comme nous pouvons le voir, la taille du problème grandit avec  $k$ . Un EKF itératif habituel contourne le problème en ne recalculant pas explicitement la valeur de  $\hat{x}_i$  pour  $i > k$  (les estimations du passé sont implicitement recalculées à la nouvelle mise à jour des mesures). Pour des modèles dynamiques et de mesures linéaires les estimations passées sont optimales, mais elles ne le sont pas pour des modèles dynamiques ou de mesures non linéaires.

### BSEKF :

Le Backward smoothing EKF traduit en EKF à Lissage rétrograde EKF [ref~Psi05](#) améliore l'EKF itératif en linéarisant un nombre fini de mesures dans le passé quand une nouvelle mesure est traitée.

$$J_k = \frac{1}{2} \sum_{i=k-m(k)}^{k-1} \{ [\tilde{y}_{i+1} - h_{i+1}(x_{i+1})]^T R_{i+1}^{-1} [\tilde{y}_{i+1} - h_{i+1}(x_{i+1})] + w_i^T Q_i^{-1} w_i \} + \frac{1}{2} (x_{k-m(k)} - \hat{x}_{k-m(k)}^*)^T P_{k-m(k)}^{*-1} (x_{k-m(k)} - \hat{x}_{k-m(k)}^*)$$

Cette équation conserve toutes les non-linéarités des itérations  $m(k)$  les plus récents via le second terme quadratique qui est une approximation de la fonction de perte  $J_{k-m(k)}$  pour  $x_{k-m(k)}$  fixe.

Une valeur  $m_{target}$  pour le nombre d'étapes à retenir est choisie pour équilibrer précision et effort de calcul.

### **Observateur non linéaire [47-48-50-51-52]**

Peut être très précis mais repose sur des observations non bruitée (mais biaisées) du vecteur d'état (un observateur est un terme utilisé pour renvoyer à des cas où on dispose d'observation parfaites du système).

### **Filtre à particules [32 33 104]**

Le plus grand avantage du filtrage à particules est que les bruits de mesure et de processus peuvent ne pas être de densité de probabilité gaussienne. Un filtre à particules est notre seul recours quand les densités de probabilité en jeu lors de l'estimation sont significativement non gaussiennes.

Cependant l'exactitude de l'estimation d'un filtre à particules dépend de la dimensionnalité des jeux de données. Pour faire simple, il faut une très grande quantité de données (typiquement, fortement sur-échantillonner notre signal avec les erreurs qui en découlent) pour pouvoir disposer de jeux de données suffisants pour implémenter un filtre à particules. Il en résulte par conséquent une complexité de calcul et des temps de traitement qui se comptent en heures et en jours plutôt qu'en secondes et en minutes. Pour ces raisons, le filtrage à particules reste limité à des utilisations dans des domaines très limités tel que le domaine spatial.

### **Méthodes QUEST (c'est le filtrage complémentaire) [32 33 104]**

Utilisent diverses hypothèses pour dériver des filtres simplifiés : sont généralement sous optimaux en comparaison avec des EKF mais utilisent des équations linéaires ou pseudo linéaires pour estimer les états d'un système non linéaire.

### **Unscented Kalman filter (UKF) [32 33 104]**

Lorsque les modèles de transition d'état et d'observation, c'est-à-dire les fonctions de prédiction et de mise à jour sont fortement non linéaires, un EKF peut donner des estimation particulièrement mauvaises car la covariance est propagée par linéarisation du modèle non linéaire sous-jacent. L'UKF utilise une technique d'échantillonnage déterministe connue sous le nom d'UT sélectionner un ensemble minimal de points d'échantillonnage (appelés points sigma) autour de la moyenne. Les points sigma sont ensuite propagés par les fonctions non linéaires, à partir desquelles une nouvelle estimation de la moyenne et de la covariance est ensuite formée. Le filtre résultant dépend de la façon dont les statistiques transformées de l'UT sont calculées et de l'ensemble des points sigma utilisés- il faut noter qu'il est toujours possible de construire de nouveaux UKF de manière cohérente. De plus, cette technique élimine la nécessité de calculer explicitement les matrices Jacobiennes, ce qui, pour les fonctions complexes, peut être une tâche difficile en soi ou impossible.

### **Comparaison des différents algorithmes pour la localisation d'un robot**

**ref** Comme le montre le tableau, dans une application analogue à la notre, l'EKF paraît être le meilleur compromis temps de calcul-performance. (SMO dans le tableau atteint de meilleurs performances que l'EKF mais c'est un observateur non linéaire.)

	RMS error [deg]	Amount of memory [M Bytes]	Convergence time [sec]	Computation time for PC [sec]	Computation time for OBC 750 [sec]
<b>Least Squares</b>	Roll [0.0726] Pitch [0.0563] Yaw [0.0250] 0.0951	0.765	60	0.56	12.30
<b>Quest Method</b>	Roll [0.0402] Pitch [0.0278] Yaw [0.0271] 0.0559	0.6484	/	0.31	6.81
<b>Optimized TRIAD</b>	Roll [0.0415] Pitch [0.0332] Yaw [0.0312] 0.0616	0.8867	/	0.49	10.76
<b>EKF</b>	Roll [0.0294] Pitch [0.0294] Yaw [0.0294] 0.0509	0.9492	100	0.74	16.25
<b>UKF</b>	Roll [0.0297] Pitch [0.0316] Yaw [0.0295] 0.0524	10.218	200	2.72	59.75
<b>SMO</b>	Roll [0.0245] Pitch [0.0192] Yaw [0.0195] 0.0367	1.4063	30	0.63	13.84

- <https://www.seas.harvard.edu/courses/cs281/papers/unscented.pdf>
- <http://iopscience.iop.org/article/10.1088/1742-6596/783/1/012017/pdf>
- <http://folk.ntnu.no/tomgra/papers/IAC2011.pdf>
- <https://pdfs.semanticscholar.org/8a00/71ca6a36789e14668162962bf8cf3a04e8dd.pdf>
- <http://www.swge.inf.br/cba2014/anais/PDF/1569926775.pdf>
- [http://www.arpnjournals.org/jeas/research\\_papers/rp\\_2016/jeas\\_0416\\_3978.pdf](http://www.arpnjournals.org/jeas/research_papers/rp_2016/jeas_0416_3978.pdf)
- <http://folk.ntnu.no/tomgra/papers/IAC2011.pdf>
- [https://radionavlab.ae.utexas.edu/images/stories/files/psiaki\\_visit/psiaki\\_JGCD2005V2\\_8No5pp885\\_894.pdf](https://radionavlab.ae.utexas.edu/images/stories/files/psiaki_visit/psiaki_JGCD2005V2_8No5pp885_894.pdf)
- <https://arxiv.org/pdf/1602.07733.pdf>
- **unscented kalman filter :**
- [https://users.aalto.fi/~ssarkka/course\\_k2009/slides\\_5.pdf](https://users.aalto.fi/~ssarkka/course_k2009/slides_5.pdf)
- <https://www.seas.harvard.edu/courses/cs281/papers/unscented.pdf>

## Annexe 4 : Illustration des ordres d'échantillonnage

Nous allons illustrer les différents ordres d'échantillonnage à travers un exemple.

### Méthode d'Euler (Ordre 1)

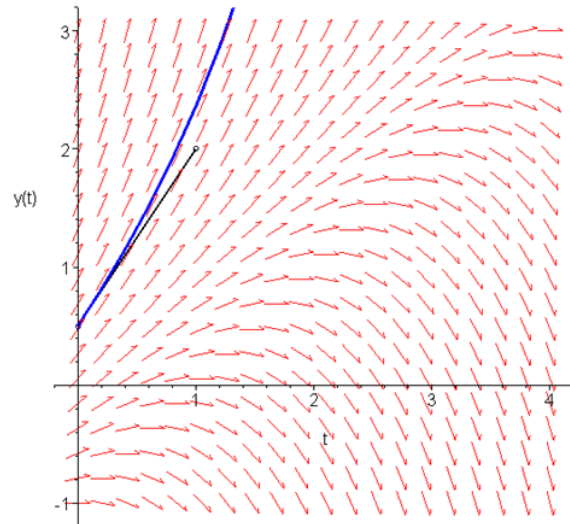
Valeur initiale  $y(0) = 0.5$ .

On essaie d'approximer  $y(1)$

**On suit la pente en  $(0, 0.5)$  sur une distance de 1.**

$$y(1) \approx 0.5 + 1.5 \cdot 1 = 2$$

(valeur réelle  $\approx 2.35914$ )



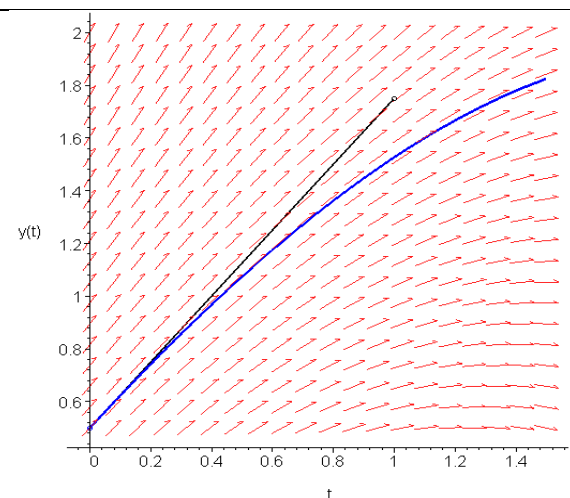
### Méthode Midpoint (Ordre 2)

Valeur initiale :  $y(0) = 0.5$

$$y^{(1)}(t) = \frac{1}{2}y(t) - t + 1$$

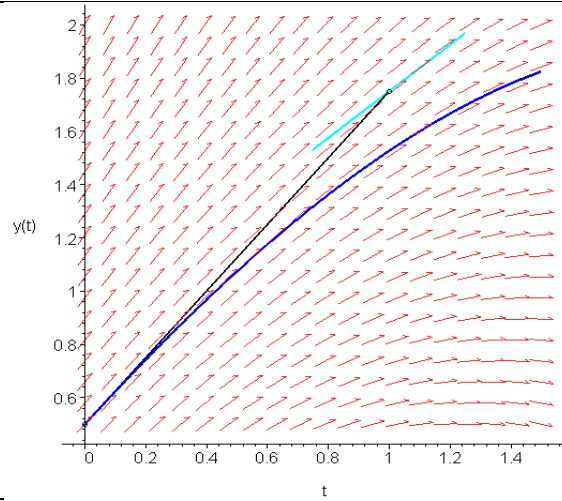
**Etape 1** : Approximation d'Euler de  $y(1)$ :

$$y(t_1) \approx 0.5 + 1 \cdot 1.25 = 1.75$$



**Etape 2 :** On échantillonne le signal à (1, 1.75), et on trouve

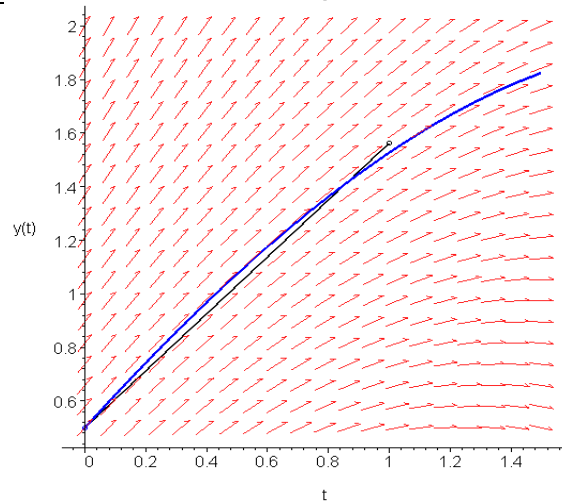
$$K_1 = f(t_1, y_1) = f(1, 1.75) = 0.875.$$



**Etape 3 :** Pente moyenne de  $\frac{1}{2}(K_0 + K_1) = 1.0625$

$$y_1 = y_0 + \frac{1}{2}(K_0 + K_1)h = 0.5 + 1.0625 = 1.5625$$

(valeur réelle  $\approx 1.5$ )



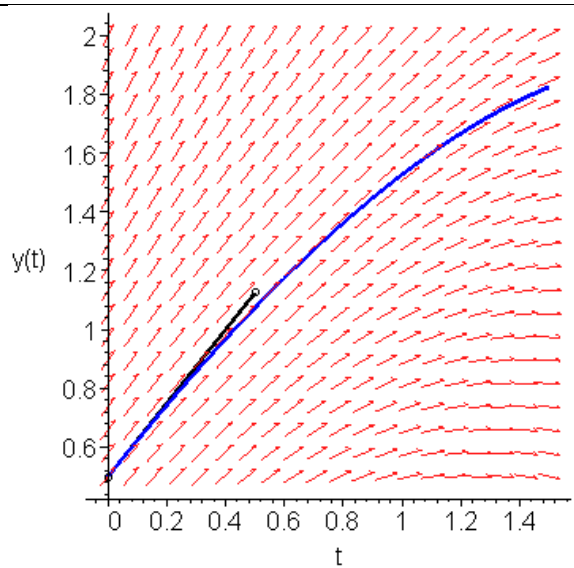
### Méthode RK4 (Ordre 4)

Posons  $h = t_1 - t_0$

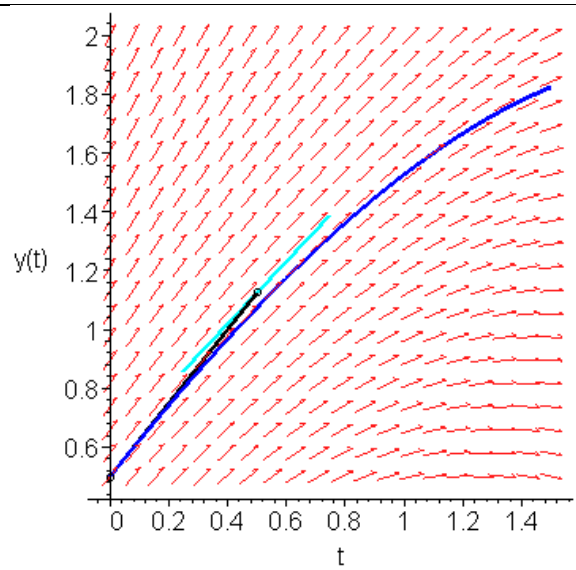
à  $(t_0, y_0) : K_0 = f(t_0, y_0)$

**Etape 1 :** On utilise cette pente pour estimer

$$y(t_0 + h/2) \approx y_0 + \frac{1}{2} h K_0$$

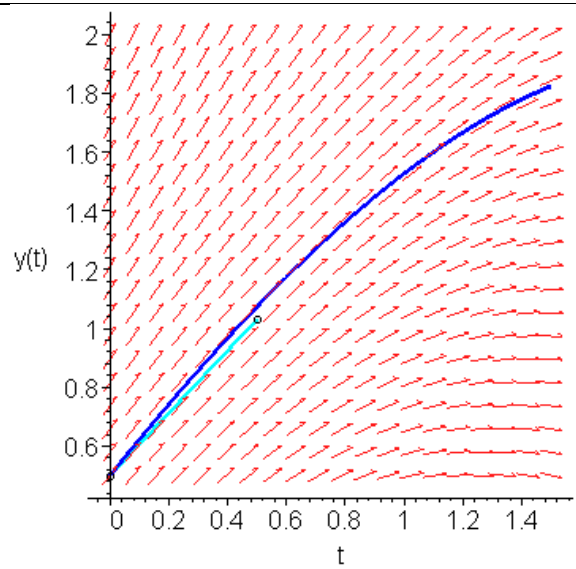


**Etape 2 :** On calcule la pente  $K_1 = f(t_0 + \frac{1}{2}h, y_0 + \frac{1}{2}h K_0)$



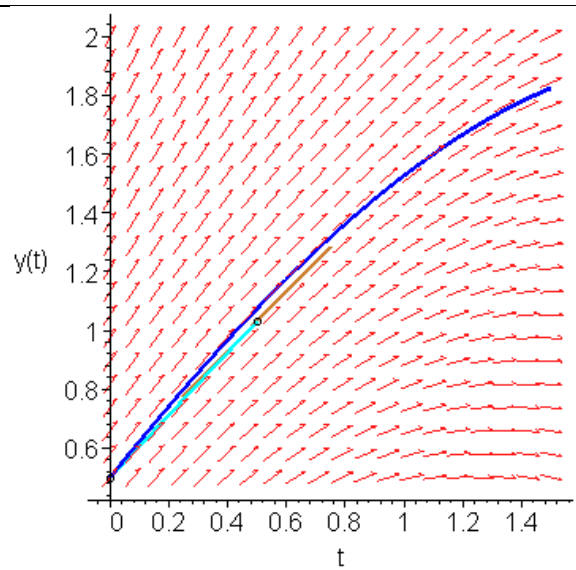
De même :

$$y(t_0 + h/2) \approx y_0 + \frac{1}{2} h K_1$$



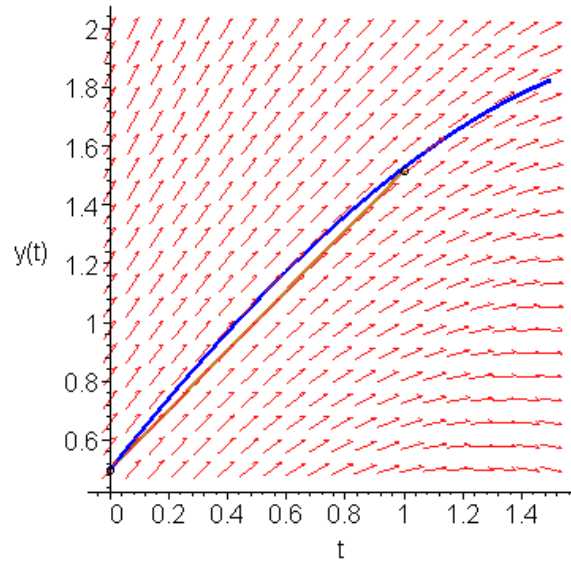
**Etape 3 :** On calcule

$$K_2 = f(t_0 + \frac{1}{2}h, y_0 + \frac{1}{2}h K_1)$$



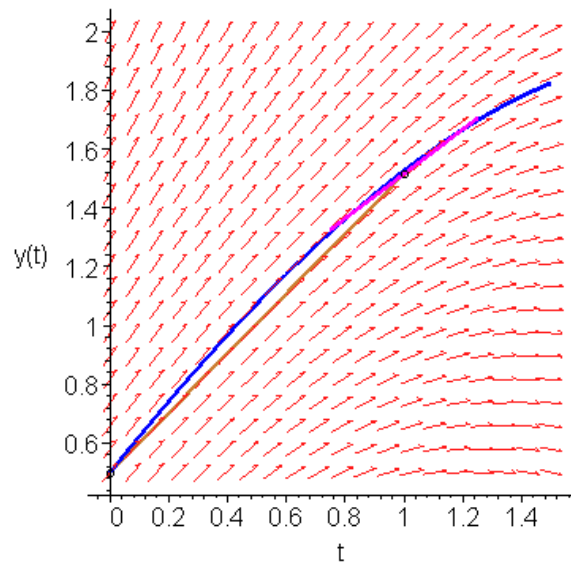
On calcule

$$y(t_1) = y(t_0 + h) \approx y_0 + hK_2$$



**Etape 4 :** On calcule :

$$K_3 = f(t_0 + h, y_0 + hK_2)$$



**Etape 5 :** On moyenne  $K_0, K_1, K_2,$  and  $K_3$   
(approximations de la pente entre  $[t_0, t_1]$ ),

