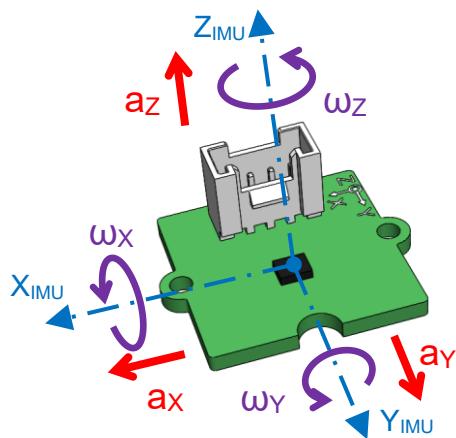


AU COEUR DES CENTRALES INERTIELLES



OBJECTIFS :

- ✓ Programmer l'acquisition des mesures d'accélérations et de vitesses angulaires avec une centrale inertie.
- ✓ Programmer les calculs de l'attitude de la centrale à partir des composantes de l'accéléromètre et du gyromètre et connaitre leurs limites.
- ✓ Programmer un filtrage simple par fusion des 2 capteurs.
- ✓ Découvrir les principes de filtrages avancés.

MATERIEL NECESSAIRE :

- ✓ Une carte Arduino avec son Shield [SeedStudio Base shield Grove](#) et un câble USB type A vers B
- ✓ Un module centrale inertie SeedStudio Grove parmi les suivants :
 - [IMU 6 DOF](#)
 - [IMU 9 DOF](#)
 - [IMU 9 DOF V2](#)
 - [IMU 10 DOF](#)
- ✓ Eventuellement un module [grove écran LCD](#)
- ✓ Les câbles Grove de connexion



[Grove IMU 6 DOF](#)



[IMU_10 DOF](#)

Préambule :

Ce document est destiné à servir de support pour une formation à la découverte de la programmation des centrales inertielles dans un environnement Arduino.

Cette formation s'adresse à des enseignants de pré baccalauréat en spécialité Sciences de l'ingénieur, en STI2D ou éventuellement en spécialité Numérique et Sciences Informatiques.

Les explications et repères historiques sont très succincts et limités à la découverte des concepts.

Les abstractions et les démonstrations mathématiques pour la compréhension des calculs peuvent être relativement complexes pour les élèves de pré-bac.

Pour des questions pédagogiques, elles font donc l'objet de vulgarisations volontaires et les détails des calculs simplifiés ne sont donnés qu'en annexe.

Le lecteur curieux pourra approfondir les différents concepts grâce aux nombreux liens revoyant vers des publications plus poussées qui sont insérés tout au long du document.

SOMMAIRE

OBJECTIFS :	0
MATERIEL NECESSAIRE :	0
1 PETITE HISTOIRE DES CENTRALES INERTIELLES :	4
2 PRINCIPES DE MESURE DE L'ACCELERATION	5
2.1 L'ACCELEROMETRE A RESSORT	5
2.2 ACCELEROMETRE PENDULAIRE A CONTRE REACTION :	6
2.3 ACCELEROMETRE A RESONANCE (OU A QUARTZ)	6
2.4 ACCELEROMETRE A CORDES VIBRANTES.....	6
2.5 L'INTEGRATION DANS UN « MEMS »	6
3 PRINCIPES DE MESURE DE LA VITESSE ANGULAIRE	7
3.1 LE GYROSCOPE MECANIQUE	7
3.2 GYROMETRE OPTIQUE	7
3.3 GYROMETRE VIBRATOIRE (MEMS)	7
4 DECOUVERTE DE LA CENTRALE	8
4.1 PRESENTATION DES MODULES IMU GROVE	8
4.2 INSTALLATION DE LA BIBLIOTHEQUE DU COMPOSANT	8
4.3 PROGRAMMER L'ACQUISITION AVEC LA BIBLIOTHEQUE.....	9
5 VERIFICATION DES VALEURS FOURNIES PAR LA CENTRALE	10
5.1 VERIFICATION DES VALEURS MESUREES PAR L'ACCELEROMETRE	10
5.2 VERIFICATION DES VALEURS MESUREES PAR LE GYROMETRE.....	11
6 COMMENT LA BIBLIOTHEQUE CALCULE LES VALEURS ?	12
6.1 LES INSTRUCTIONS AVANCEES DE LA BIBLIOTHEQUE	12
6.2 LES REGISTRES.....	12
6.3 VERIFICATION DES MESURES CALCULEES PAR LA BIBLIOTHEQUE	14
6.4 NECESSITE D'UN ETALONNAGE.....	15
7 ESTIMATION DE L'ATTITUDE	16
7.1 LE PROBLEME DE WAHBA.....	16
7.2 LES DIFFERENTES METHODES POUR L'ORIENTATION D'UN SOLIDE DANS L'ESPACE	16
7.3 PARAMETRAGE DE L'ORIENTATION AVEC LES ANGLES D'EULER.....	17
7.4 PARAMETRAGE DE L'ORIENTATION AVEC LES ANGLES DE « CARDAN »	17
7.5 PARAMETRAGE DES ANGLES DE ROULIS ET TANGAGE	18
8 CALCUL DES ANGLES AVEC L'ACCELEROMETRE	20
8.1 CHOIX N°1 : ROULIS PUIS TANGAGE	20
8.2 CHOIX N°2 : TANGAGE PUIS ROULIS.....	20
8.3 CHOIX N°3 : TANGAGE ET ROULIS INDEPENDANTS.....	20
8.4 LA FONCTION ATAN2 POUR REDUIRE LES LIMITES DE L'ARC TANGENTE	21
8.5 PROGRAMMATION DU CALCUL DES ANGLES AVEC L'ACCELEROMETRE	22
9 CALCUL DES ANGLES AVEC LE GYROMETRE	24
9.1 DE LA VITESSE A L'ANGLE.....	24
9.2 NECESSITE D'UNE INTEGRATION NUMERIQUE	24
9.3 CALCUL NUMERIQUE DES ANGLES A PARTIR DE LA VITESSE ANGULAIRE.....	24
9.4 PROGRAMMATION DU CALCUL DES ANGLES AVEC LE GYROMETRE.....	25
9.5 NECESSITE DE PRENDRE EN COMPTE L'INCLINAISON POUR LE CALCUL	27

10 VALIDITE DES ANGLES CALCULES.....	29
10.1 VALIDITE DES ANGLES CALCULES EN STATIQUE.....	29
10.2 VALIDITE DE L'ANGLE DE LACET	29
10.3 NECESSITE D'UNE PERIODE DE CALCUL PRECISE.....	29
10.4 NECESSITE D'UNE MODULATION DE L'ANGLE	30
10.5 LE PROBLEME DE LA STABILITE DES MESURES DANS LE TEMPS	30
10.6 VALIDITE DES ANGLES CALCULES EN DYNAMIQUE	30
11 LA FUSION DES CAPTEURS	31
11.1 LES PROBLEMES DE MESURE.....	31
11.2 LA NECESSITE DU FILTRAGE OU DE LA FUSION DE DONNEES	32
11.3 LE FILTRE COMPLEMENTAIRE	33
11.4 L'ALGORITHME DE FILTRAGE DE MADGWICK	35
11.5 LE FILTRE DE KALMAN	36
12 AHRS: ATTITUDE AND HEADING REFERENCE SYSTEMS.....	37
13 LE BLOCAGE DE CARDAN.....	38
13.1 UTILISATION DES QUATERNIONS.....	38
14 CONCLUSION	41
15 LES REGISTRES DE REGLAGES DE LA CENTRALE	42
15.1 AXES DE LA CENTRALE	42
15.2 CARACTERISTIQUES MECANIQUES.....	42
15.3 LES FONCTIONNALITES.....	43
15.4 LES PRINCIPAUX REGISTRES	43
16 LECTURE DES VALEURS AVEC LES REGISTRES.....	44
16.1 COMMUNICATION I2C AVEC LA CENTRALE	44
16.2 REGLAGES DE LA CENTRALE PAR LES REGISTRES	45
16.3 LECTURE DES VALEURS BRUTES	46
17 ANNEXE 1 :REGISTRES DE REGLAGES DU LSM6DS3	47
18 ANNEXE 2 : DETAILS DES CALCULS AVEC L'ACCELEROMETRE.....	50
18.1 CHOIX N°1 : ROTATION DE ROULIS PUIS DE TANGAGE.....	51
18.2 CHOIX N°2 : ROTATION DE TANGAGE PUIS DE ROULIS.....	53
19 ANNEXE 3 : DETAILS DES CALCULS AVEC LE GYROMETRE	56
20 ANNEXE 4 : LA FONCTION NUMERIQUE ATAN2.....	59
21 ANNEXE 5 : LISTE DES FICHIERS ASSOCIES	60
21.1 LISTE DES PROGRAMMES ARDUINO FOURNIS POUR LES EXPERIMENTATIONS :	60
21.2 FICHIERS POUR LES BIBLIOTHEQUES.....	60

1 PETITE HISTOIRE DES CENTRALES INERTIELLES :

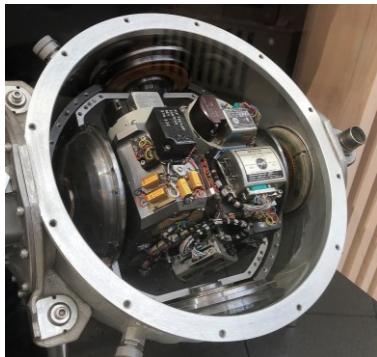
Les centrales inertielles sont utilisées pour mesurer l'**orientation** d'un système dans l'espace **à partir de son mouvement**.

Les principes physiques mis en œuvre sont connus depuis fort longtemps :

- ✓ Machine de Bohnenberger (début 19^{ème} siècle) à 3 **cardans** qui a permis de mettre en évidence les effets de **précession** et l'effet **gyroscopique** ;
- ✓ L'invention du principe du **gyroscope** par Ernest Lamarle et sa mise au point par Léon Foucault en 1852 ;
- ✓ L'invention du **gyrocompas**, dont les premiers brevets, de systèmes commercialisés, sont déposés en 1904 ;
- ✓ Puis l'émergence de la **navigation à l'estime**, l'**horizon artificiel** et enfin la **navigation inertielle**.



[The Machine of Bohnenberger](#)
Début 19^{ème}

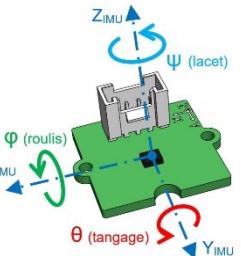


[Apollo Inertial Measurement](#)

La navigation inertielle est documentée à partir des années 1940 pour le guidage analogique du missile Allemand V-2 pendant la seconde guerre mondiale puis dans les recherches sur les missiles balistiques comme le projet Jupiter aux USA dans les années 1950.

C'est en 1961, à l'occasion du **programme Apollo** que la NASA, en partenariat avec le MIT, a mis au point le premier ordinateur embarqué : l'Apollo primary guidance, navigation, and control system (PGNCS).

C'est le premier ordinateur à circuit intégré équipé d'un **IMU (inertial measurement unit)** monté sur cardan à 3 axes.



Comme pour nos centrales actuelles il était composé de :

- ✓ 3 **gyromètres** : Rate integrating gyroscope (RIG) qui permettent de mesurer les **vitesses angulaires** autour des 3 axes.
- ✓ 3 **accéléromètres** : Pulsed Integrating Pendulous Accelerometer (PIPA ou PIGA), qui permettent de mesurer les **accélérations** linéaires suivant les 3 axes.

C'est grâce à ces 6 paramètres que l'ordinateur de bord est capable d'estimer l'orientation de la navette spatiale suivant les 3 angles de Roulis, Tangage et Lacet.



[Gyroscope JG7005](#)
Années 1950

L'innovation incrémentale a permis, par la suite, d'optimiser puis de miniaturiser ces instruments de mesures pour les faire passer de modules électromécaniques de taille et de masse imposantes à des **microsystèmes électromécaniques (mems)** de quelques mm³ de volume.

gyropodes...



[Centrale SBG Elypse AHRS](#)

Ce sont ces composants qui sont aujourd'hui implantés dans les moyens de transports aériens nécessitant une stabilisation active mais aussi intégrés dans tous les systèmes grand public tels que les smartphones, les montres connectées, les manettes de jeux vidéo, les drones, les

Time and navigation – the untold story of getting from here to where /www.ion.org/museum/

An historical perspective on inertial navigation systems - Daniel Tazarte max.q - L'aventure apollo - Guidance, Navigation and Control System

cadden.fr - La centrale inertielle

De la Terre à la Lune. Apollo un formidable voyage - LES ÉQUIPEMENTS DU PGNCS

2 PRINCIPES DE MESURE DE L'ACCELERATION

2.1 L'accéléromètre à ressort

La mesure de l'accélération, se base sur la mesure du déplacement **relatif** d'une **masse sismique** par rapport à un **châssis mobile**.

La « masse sismique » est liée de manière élastique au châssis de telle sorte qu'elle soit en équilibre en position initiale lorsqu'elle est immobile.

La valeur de l'accélération sera calculée à partir de la **mesure physique du déplacement** de la masse sismique par rapport au châssis.

En effet, lors d'une accélération axiale, du fait de son inertie propre, la masse sismique a tendance à rester dans sa position et un déplacement relatif x apparaît entre la masse et le châssis :

- ✓ Accélération positive = déplacement négatif.
- ✓ Accélération négative = déplacement positif.

Le calcul repose sur les lois de la dynamique (2ème loi de Newton) :

$$\sum \vec{F}_{\text{Ext}} = m \cdot \vec{a}$$

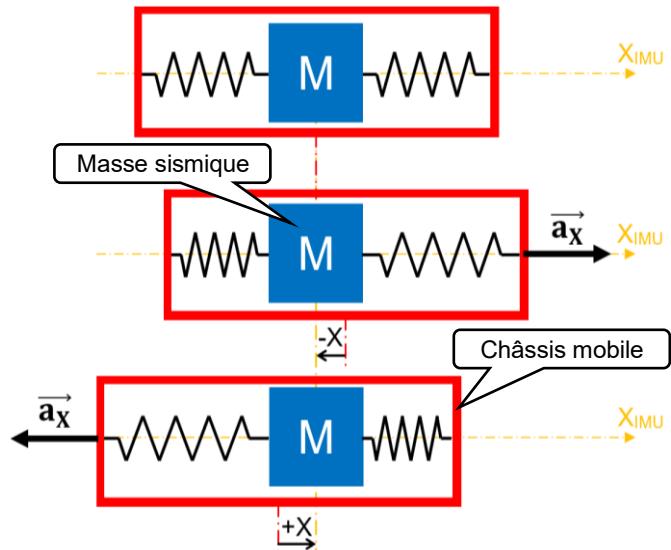
En étudiant les composantes sur l'axe horizontal :

$$-F_{\text{RESSORT}} = m \cdot a_x$$

- $F_{\text{RESSORT}} = K \cdot x$
- K : raideur du ressort
- m : masse

L'accélération est donc fonction du déplacement :

$$a_x = \frac{-K \cdot x}{m}$$



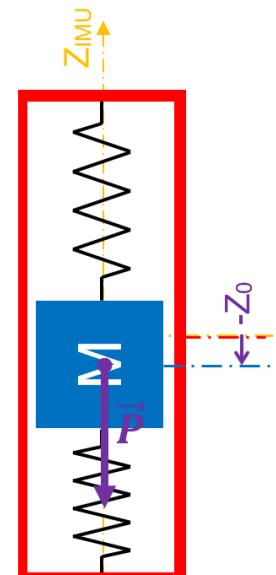
2.1.1 Le problème de l'accélération verticale

Pour l'accélération verticale, la gravité vient « perturber » la mesure puisque la masse sismique est, **EN STATIQUE**, naturellement **décalée vers le bas**.

Ce **décalage négatif** Z_0 induit une valeur de mesure de **l'accélération verticale POSITIVE** égale à la valeur de l'accélération gravitationnelle.

C'est pour cette raison, qu'au repos, l'accéléromètre renvoie toujours une valeur d'1g (9.81 m/s²), **POSITIVE** et purement verticale.

Et c'est aussi grâce à ce phénomène qu'il est possible de retrouver des angles d'inclinaison du capteur en Statique.

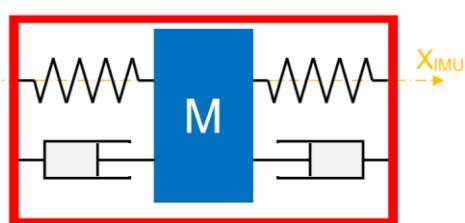


2.1.2 Le ressort amorti

En réalité le principe est un peu plus complexe puisque le système sismique peut être modélisé comme un système amorti et donc le calcul de l'accélération devra tenir compte du coefficient d'amortissement et de la vitesse de la masse :

$$-F_{\text{RESSORT}} = m \cdot a_x + c \cdot v_x + K \cdot x$$

La qualité de la mesure dépend de la taille de la masse, de l'amortissement et de la constante de rappel du ressort.



2.2 Accéléromètre pendulaire à contre réaction :

Le fonctionnement repose sur un rotor inertiel entraîné en rotation autour d'un axe et monté sur un cardan de manière asymétrique.

Le cardan est en pivot suivant l'axe de mesure de l'accélération, sa rotation est contrôlée par un moteur.

L'accélération entraîne un couple de précession sur l'axe de rotation du cardan.

La mesure de l'accélération sera proportionnelle au courant nécessaire pour repositionner le cardan dans sa position d'origine.

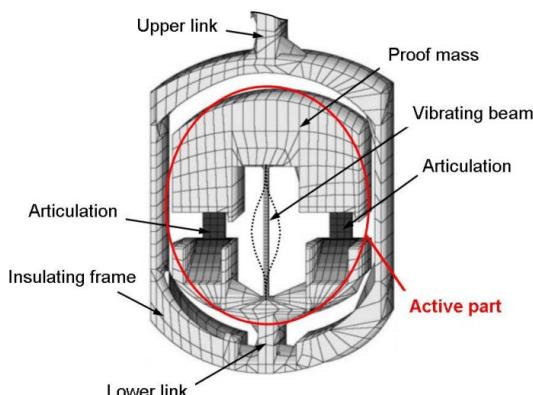
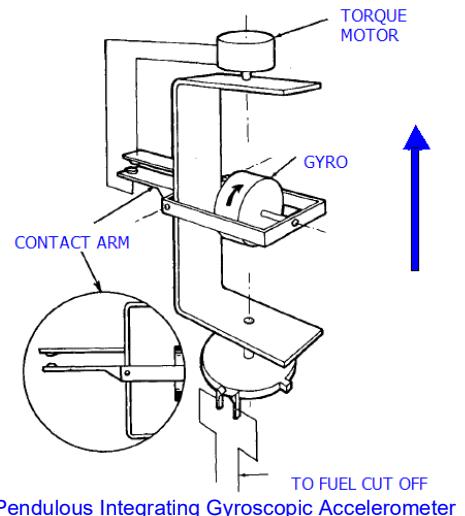


Fig. 2. VIA transducer.
[A New Quartz Monolithic Differential Vibrating Beam Accelerometer](#)

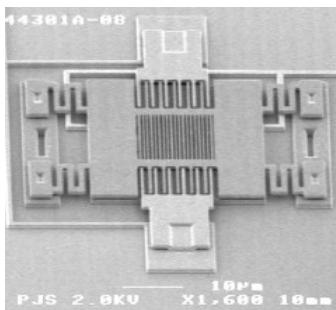
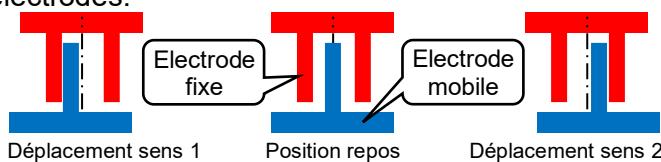
2.4 Accéléromètre à cordes vibrantes

Le principe est proche de l'accéléromètre à résonance et repose sur la mesure de la variation de fréquence des cordes lorsque leurs longueurs sont modifiées par le déplacement de la masse.

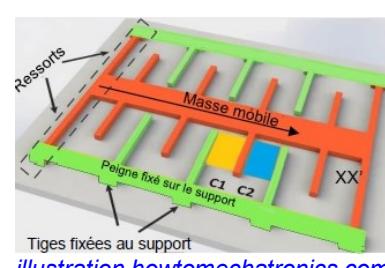
2.5 L'intégration dans un « mems »

Les circuits intégrés de type mems utilisés dans les centrales inertielles sont fabriqués à base de matériaux semi-conducteurs sous forme de « peignes » entremêlés dont l'un est fixe et l'autre peut se déplacer de manière élastique par rapport à la partie fixe.

Les « doigts » des peignes sont en réalité des électrodes électriques dont la variation de « capacitance » évolue en fonction de la distance entre les électrodes.



Pour l'accélération le principe repose sur la mesure du déplacement de l'électrode mobile lors de la mise en mouvement.



[illustration howtomechatronics.com](#)

3 PRINCIPES DE MESURE DE LA VITESSE ANGULAIRE

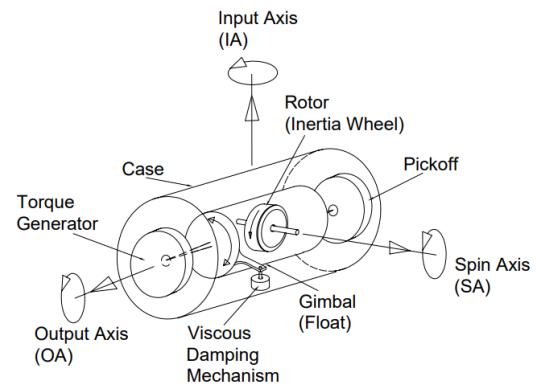
3.1 Le gyroscope mécanique

Un gyromètre mécanique est composé d'un rotor inertiel tournant à l'intérieur d'une structure à cardan 3 axes.

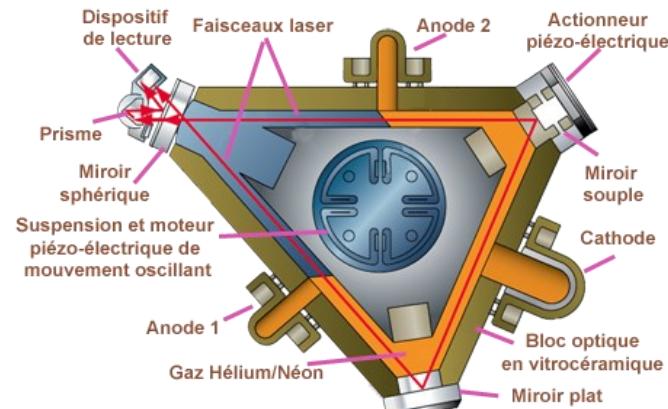
Il utilise le principe de la conservation du moment angulaire qui conserve son orientation globale dans l'espace.

Il est de ce fait possible de mesurer l'angle relatif entre les cardans grâce à des capteurs de rotation et en déduire une vitesse angulaire.

L'inconvénient vient des frictions mécaniques internes qui induisent une dérive dans le temps.



[Illustration USD Dynamic](#)



Source: www.faa.gov/handbooks_manuals/media

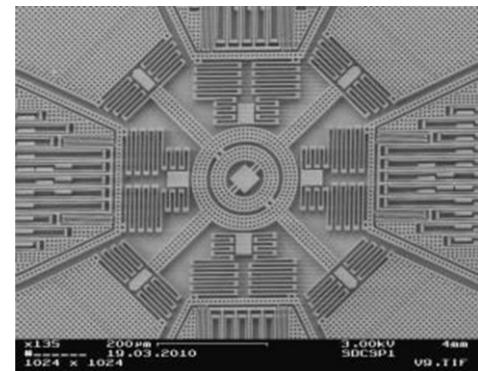
[Illustration l'avionnaire.fr](#)

3.2 Gyromètre optique

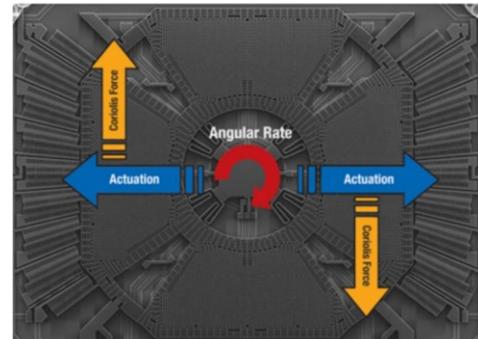
Les gyromètres optiques utilisent deux faisceaux lumineux parcourant le même trajet, mais dans des directions opposées.

Un détecteur de phase mesure, par interférométrie, un changement relatif de phase occasionné par l'effet Sagnac lorsque le système est soumis à une rotation.

Il existe deux types de gyroscope à laser : Ring Laser Gyroscope (RLG) et Fiber-Optic Gyroscope (FOG).



[Illustration ENS Cachan](#)



[Illustration strumentazioneelettronica.it](#)

[Thèse de Maîtrise en sciences géomatiques de Hickey, Jean-René](#)

4 DECOUVERTE DE LA CENTRALE

4.1 Présentation des modules IMU GROVE

Il existe une multitude de centrales inertielles de type mems commercialisées par différents constructeurs et qui possèdent tous, au moins, un gyromètre 3 axes et un accéléromètre 3 axes.

Certaines puces peuvent aussi intégrer d'autres systèmes de mesures comme un **magnétomètre**, un **compas** ou un **baromètre** permettant de réaliser des mesures plus poussées.

Cette formation propose de s'intéresser à l'acquisition et au calcul d'attitude à partir des seules mesures des accélérations et vitesses angulaires.

Le matériel proposé se limite aux modules Seedstudio de type Grove pour faciliter les connexions mais vous serez à même de pouvoir adapter les notions à d'autres capteurs concurrents.

Voici un tableau présentant les différents modules Seedstudio que l'on retrouve facilement dans le commerce avec leurs composants et les liens vers les **documentations techniques (datasheet)** :

Module Grove	IMU 6 DOF	IMU 9 DOF	IMU 9 DOF V2	IMU 10 DOF
Composant principal	LSM6DS3	lcm20600	InvenSense TDK MPU 9250	
Accéléromètre.	$\pm 2 \text{ g} ; \pm 4 \text{ g} ; \pm 8 \text{ g} ; \pm 16 \text{ g}$			
Gyromètre	$\pm 125 ; 245 ; 500 ; 1000 ; 2000 \text{ }^{\circ}/\text{s}$		$\pm 250 \text{ }^{\circ}/\text{s} ; \pm 500 \text{ }^{\circ}/\text{s} ; \pm 1000 \text{ }^{\circ}/\text{s} ; \pm 2000 \text{ }^{\circ}/\text{s}$	
Magnétomètre			res 14 bit (0.6 μ T/LSB) range $\pm 4800\mu\text{T}$	
Autre mesure	AKM AK09918 Compas		Bosch BMP280 baromètre	

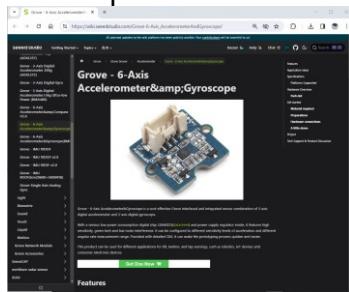
4.2 Installation de la bibliothèque du composant

Le premier réflexe lors de l'utilisation d'un nouveau composant est d'utiliser les ressources fournies par les « constructeurs ». Pour notre cas il suffit de se rendre sur la page wiki Seedstudio du module.

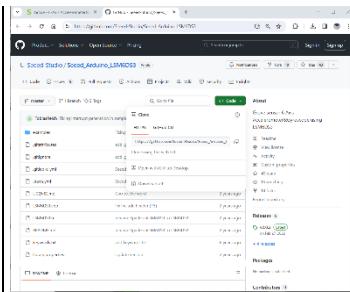
La page nous donne accès à la bibliothèque du composant :

- ✓ [Seeed Arduino LSM6DS3-master.zip](#) pour la puce LSM6DS3
- ✓ [Grove IMU_9DOF-master.zip](#) pour la puce MPU9250

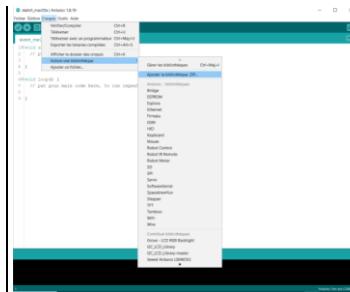
Après l'installation de la dite bibliothèque il est possible d'ouvrir les programmes d'exemples et de les tester rapidement en les téléversant sur l'arduino.



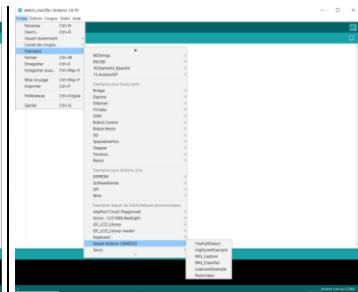
1 accéder au wiki du composant



2 télécharger la bibliothèque



3 installer la bibliothèque sur le logiciel arduino



4 accéder aux exemples de la bibliothèque

4.3 Programmer l'acquisition avec la bibliothèque

A partir des programmes d'exemples il est assez simple de comprendre les instructions permettant de récupérer et d'afficher les valeurs lues par la centrale.

Par exemple pour la centrale LSM6DS3 on peut extraire les principales instructions suivantes :

```
#include "LSM6DS3.h"
#include "Wire.h"                                     Importation de la bibliothèque dans le programme C

LSM6DS3 myIMU(I2C_MODE, 0x6A);                   Importation de « wire.h » pour la communication I2C

myIMU.begin()                                      Déclaration de la méthode « myIMU » pour l'appel des fonctions

myIMU.readRawAccelX()
myIMU.readRawAccelX()
myIMU.readRawAccelX()
myIMU.readRawGyroX()
myIMU.readRawGyroY()
myIMU.readRawGyroZ()                                Démarrage de la centrale

myIMU.readFloatAccelX()
myIMU.readFloatAccelY()
myIMU.readFloatAccelZ()
myIMU.readFloatGyroX()
myIMU.readFloatGyroY()
myIMU.readFloatGyroZ()                            Lecture des valeurs brutes

myIMU.readTempC()
myIMU.readTempF()                                  Récupération des valeurs calculées

myIMU.readTempC()                                  Mesure de la T° de l'IMU.
```

Pour simplifier votre première expérience, voici un [premier programme](#) permettant de récupérer et d'afficher les valeurs brutes et calculées de la centrale.

```
#include "LSM6DS3.h"
#include "Wire.h"

LSM6DS3 myIMU(I2C_MODE, 0x6A);
```

Remarques : Afin de lire les valeurs vous devez utiliser le moniteur série du logiciel arduino. Il sera réglé ici à la fréquence de 115200 bauds.

```
int16_t ax_brut, ay_brut, az_brut, gx_brut, gy_brut, gz_brut;

void setup() {
    Serial.begin(115200);
    if (myIMU.begin() != 0) {
        Serial.println("Problème de connexion avec la centrale");
    } else {
        Serial.println("Centrale prête à fonctionner ! ");
    }
}

void loop() {
    ax_brut = myIMU.readFloatAccelX();
    ay_brut = myIMU.readFloatAccelY();
    az_brut = myIMU.readFloatAccelZ();
    gx_brut = myIMU.readFloatGyroX();
    gy_brut = myIMU.readFloatGyroY();
    gz_brut = myIMU.readFloatGyroZ();

    float ax_biblio = myIMU.readFloatAccelX();
    float ay_biblio = myIMU.readFloatAccelY();
    float az_biblio = myIMU.readFloatAccelZ();
    float gx_biblio = myIMU.readFloatGyroX();
    float gy_biblio = myIMU.readFloatGyroY();
    float gz_biblio = myIMU.readFloatGyroZ();

    Serial.println(" -- Accélémètre : --");
    Serial.print("X_raw="); Serial.print(ax_brut); Serial.print("-X_réel="); Serial.print(ax_biblio,2); Serial.println("g");
    Serial.print("Y_raw="); Serial.print(ay_brut); Serial.print("-Y_réel="); Serial.print(ay_biblio,2); Serial.println("g");
    Serial.print("Z_raw="); Serial.print(az_brut); Serial.print("-Z_réel="); Serial.print(az_biblio,2); Serial.println("g");
    Serial.println(" -- Gyromètre : --");
    Serial.print("X_raw="); Serial.print(gx_brut); Serial.print("-X_réel="); Serial.print(gx_biblio,2); Serial.println(" °/S");
    Serial.print("Y_raw="); Serial.print(gy_brut); Serial.print("-Y_réel="); Serial.print(gy_biblio,2); Serial.println(" °/S");
    Serial.print("Z_raw="); Serial.print(gz_brut); Serial.print("-Z_réel="); Serial.print(gz_biblio,2); Serial.println(" °/S");
    Serial.println();
    delay(500);
}
```

5 VERIFICATION DES VALEURS FOURNIES PAR LA CENTRALE

5.1 Vérification des valeurs mesurées par l'accéléromètre

Pour l'accéléromètre, rien de plus simple, en statique il mesure directement la valeur de l'accélération gravitationnelle.

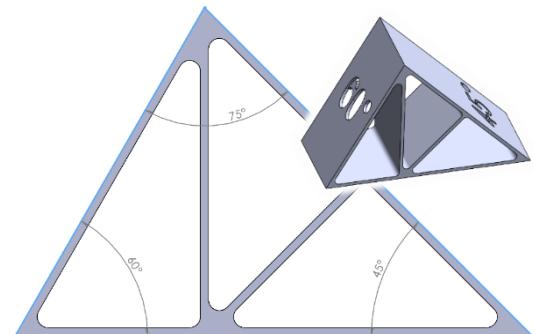
Il suffit donc d'interpréter les valeurs des 3 composantes.

Une cale pentée est proposée pour positionner le capteur :



- ✓ à la verticale,
- ✓ incliné avec un angle de 45°, 60° ou 75°.

Le fichier est disponible [sur le dépôt github du gyroscope](#) au format « .stl » et imprimable en 3D rapidement.



A FAIRE: Vérifier les valeurs réelles calculées par l'accéléromètre pour différentes positions et compléter le tableau de vérification (page suivante).

Ax				
Ay				
Az				

Attention l'accéléromètre mesure l'inverse de la gravité ([voir problème de l'accélération verticale](#)).

A FAIRE: Poser la centrale sur la table et secouez-la vigoureusement suivant un des axes horizontaux.

A FAIRE: Dans la liste des valeurs affichées dans le moniteur série, relevez les valeurs minimales et maximales.

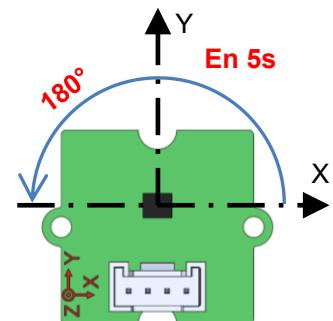
A FAIRE: En déduire la valeur pleine échelle réglé par défaut dans la bibliothèque.

5.2 Vérification des valeurs mesurées par le gyromètre

Le but est de valider la valeur de vitesse angulaire fournie par le gyromètre. Il est donc nécessaire de faire tourner la centrale autour d'un axe rotation à vitesse stabilisée et connue.

Une première approche très simplifiée consiste à :

1. Poser la centrale sur la table,
2. Effectuer une rotation à vitesse la plus régulière possible,
3. Observer la valeur « moyenne » de la vitesse autour de l'axe Z,
4. Vérifier la valeur par calcul : $\omega_Z = \frac{180}{5} = 36^\circ/\text{s}$



A FAIRE: Poser la platine à plat en statique et vérifier les valeurs fournies, que constatez-vous ?

A FAIRE: En gardant la platine à plat, effectuer une rotation autour de l'axe z, avec une vitesse la plus constante possible et vérifier que seule la valeur de la composante sur z de la vitesse angulaire change.

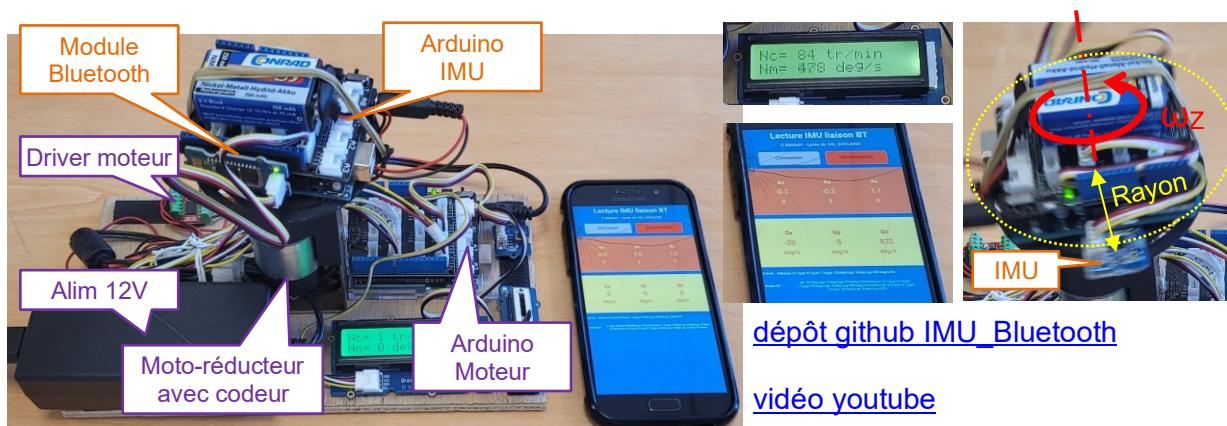
A FAIRE: Vérifier que la valeur mesurée soit cohérente.

A FAIRE: Essayer de retrouver la valeur pleine échelle par défaut du gyromètre.

Une solution plus précise consiste à utiliser un banc moteur comme dans l'exemple proposé d'un motoréducteur avec codeur intégré piloté en fréquence de 0 à 500 tr/min.

Le gyromètre possède une « valeur pleine échelle » paramétrable entre 125 et 2000 °/s ([voir caractéristiques de la centrale](#)) soit des fréquences **jusqu'à 333 tr/min**.

Pour la lecture des informations du codeur, il est possible d'utiliser un **collecteur tournant à au moins 4 canneaux** pour une communication « filaire » ou, comme ici, une centrale autonome qui communique en Bluetooth avec un smartphone muni d'une application adhoc.



6 COMMENT LA BIBLIOTHEQUE CALCULE LES VALEURS ?

6.1 Les instructions avancées de la bibliothèque

En étudiant un peu plus en détail les différents exemples de la bibliothèque on se retrouve rapidement avec des instructions étranges :

```
uint8_t dataToWrite = 0;
```

Déclaration d'une variable entière non signée 8 bits

```
myIMU.readRegister(&dataToWrite, LSM6DS3_ACC_GYRO_CTRL4_C);
```

Lecture d'un **registre** 8 bits

```
myIMU.readRegisterInt16(&dataToWrite, LSM6DS3_ACC_GYRO_OUTX_L_XL);
```

Lecture d'un registre 16 bits

```
dataToWrite |= LSM6DS3_ACC_GYRO_BW_XL_100Hz;
dataToWrite |= LSM6DS3_ACC_GYRO_FS_XL_8g;
dataToWrite |= LSM6DS3_ACC_GYRO_ODR_XL_104Hz;
```

Création d'une valeur binaire par addition de bits à partir de mots clés spécifiques

```
myIMU.writeRegister(LSM6DS3_ACC_GYRO_CTRL1_XL, dataToWrite)
```

Ecriture sur un registre 8 bits

Pour un programmeur initié aux microprocesseurs, les **registres** sont des notions évidentes et comme les centrales ne sont en réalité que des microcontrôleurs spécialisés intégrant des technologies mems, ils possèdent aussi leurs propres registres qu'il est possible de lire ou d'écrire en passant par les protocoles de communication.

La communication entre l'arduino et l'IMU passe par une liaison de type série, nous utiliserons la liaison **i2C** qui est facilement exploitable.

6.2 Les REGISTRES

Un registre est en réalité une mémoire interne au microcontrôleur qui peut être de taille variable entre 8 et 64 bits voire plus pour certains processeurs modernes.

Chaque registre correspond donc à **valeur**, codée ici sur **1 octet** donc 8 bits et possède une **adresse** afin de savoir comment la lire ou écrire.

La lecture de la documentation technique (datasheet) est un **préalable obligatoire** pour connaître l'utilité des différents **registres**, leurs **adresses** mais aussi les caractéristiques et les possibilités de réglages de chaque centrale.

La lecture de ce document n'est pas des plus digestes mais sera un passage obligé pour connaître tous les détails et les possibilités offertes par les centrales.

La table des registres permet de visualiser la liste et les adresses et indique qu'il existe 2 types de registres :

- ✓ Les registres de lecture des valeurs brutes
- ✓ Les registres de réglages.

ST life.augmented **LSM6DS3**

iNEMO inertial module:
always-on 3D accelerometer and 3D gyroscope
Datasheet - production data

Description
The LSM6DS3 is a system-in-package featuring a 3D digital accelerometer and a 3D digital gyroscope performing at 1.25 mA (up to 1.6 kHz ODR) in high-performance mode and enabling always-on low-power features for an optimal motion experience for the consumer.
The LSM6DS3 supports main OS requirements, offering real, virtual and batch sensors with 8 kbyte for dynamic data batching.

Features

- Power consumption: 0.9 mA in combo normal mode and 1.25 mA in combo high-performance mode up to 1.6 kHz.
- "Always-on" experience with low power consumption for both accelerometer and gyroscope
- Smart FIFO up to 8 kbyte based on features set
- Compliant with Android K and L
- Hard, soft ironing for external magnetic sensor corrections
- ±2/±4/±8/±16 g full scale
- ±125/±245/±500/±1000/±2000 dps full scale
- Analog supply voltage: 1.71 V to 3.6 V
- Independent I₂C supply (1.62 V)
- Compact footprint, 2.5 mm x 3 mm x 0.83 mm
- SPI/I₂C serial interface with main processor data synchronization feature
- Embedded temperature sensor
- ECOPACK®, RoHS and "Green" compliant

Applications

- Pedometer, step detector and step counter
- Significant motion and tilt functions
- Indoor navigation
- Tap and double-tap detection
- IoT and connected devices
- Intelligent power saving for handheld devices
- Vibration monitoring and compensation
- Free-fall detection
- 6D orientation detection

Part number	Temperature range [°C]	Package	Packing
LSM6DS3	-40 to +85	LGA-14L (2.5 x 3 x 0.83 mm)	Tray
LSM6DS3TR	-40 to +85	(2.5 x 3 x 0.83 mm)	Tape & Reel

Table 1. Device summary

6.2.1 Les registres de lecture des valeurs brutes

Les valeurs mesurées par la centrale sont codées sur **16 bits**. Il convient donc **lire 2 registres différents** :

- ✓ L : **registre de poids faible**
- ✓ H : **registre de poids fort**

Afin d'obtenir la valeur mesurée il est donc aussi nécessaire de réaliser une **addition binaire** pour connaître la valeur brute sur chaque axe.

Il y a donc 12 octets différents à lire pour récupérer les valeurs d'accélération et de vitesse angulaires sur chacun des 3 axes (# : axe X, You Z) :

- ✓ OUT#_L_G et OUT#_H_G pour le gyromètre adresses 22h à 27h.
- ✓ OUT#_L_XL et OUT#_H_XL pour l'accéléromètre adresses 28h à 2Dh.
- ✓ OUT_TEMP_L et OUT_TEMP_H pour la température adresse 20h et 21h.

En effet l'IMU possède aussi un capteur interne de température puisque les valeurs dépendent de celle-ci.

Name	Type	Register address		Default	Comment
		Hex	Binary		
OUT_TEMP_L	r	20	00100000	output	Temperature output data register
OUT_TEMP_H	r	21	00100001	output	
OUTX_L_G	r	22	00100010	output	
OUTX_H_G	r	23	00100011	output	
OUTY_L_G	r	24	00100100	output	
OUTY_H_G	r	25	00100101	output	
OUTZ_L_G	r	26	00100110	output	
OUTZ_H_G	r	27	00100111	output	
OUTX_L_XL	r	28	00101000	output	
OUTX_H_XL	r	29	00101001	output	
OUTY_L_XL	r	2A	00101010	output	Gyroscope output register
OUTY_H_XL	r	2B	00101011	output	
OUTZ_L_XL	r	2C	00101100	output	
OUTZ_H_XL	r	2D	00101101	output	
					Accelerometer output register

Remarque : Dans la centrale LSM6DS3, un réglage du **boutisme** appelé **Big Little Endian (BLE)** permet d'inverser l'ordre des 2 octets.

Ce réglage est utilisé dans les programmes d'exemples avancés pour être en cohérence avec les IMU de chez TDK Invensense qui utilisent l'ordre inverse ! (voir réglages registre CTRL3_C)

6.2.2 Calcul des valeurs réelles avec la sensibilité

Une fois les valeurs brutes extraites, il reste à déterminer la valeur réelle de mesure.

Pour cela on utilise la **sensibilité** du capteur qui est fonction de la valeur pleine échelle réglée :

En considérant que les mesures du capteur sont **LINEAIRES**, il est possible de calculer la sensibilité à partir de la valeur pleine échelle en **mg/LSB** ou **mdps/LSB** et du nombre n de bits du codage de la valeur brute.

Exemple pour l'accéléromètre réglé à ± 8 g :

$$\text{Sensibilité} = \frac{2 \times 8}{2^{16}} = \frac{16}{65536} = 0,0002441 \text{ g/LSB}$$

Soit 0,244 mdps/LSB

Exemple pour le gyromètre réglé à ± 500 °/s :

$$\text{Sensibilité} = \frac{2 \times 500}{2^{16}} = \frac{1000}{65536} = 0,015259 \text{ °/s/LSB}$$

Soit 15,26 mdps/LSB

$$\text{Valeur réelle} = \text{Valeur brute} \times \text{sensibilité}$$

$$\text{Sensibilité} = \frac{2 \times \text{valeur pleine échelle}}{2^n}$$

Parameter	Test conditions	Min.	Typ. ⁽¹⁾	Max.	Unit
Linear acceleration sensitivity	FS = ± 2		0.061		mg/LSB
	FS = ± 4		0.122		
	FS = ± 8		0.244		
	FS = ± 16		0.488		
Angular rate sensitivity	FS = ± 125		4.375		mdps/LSB
	FS = ± 245		8.75		
	FS = ± 500		17.50		
	FS = ± 1000		35		
	FS = ± 2000		70		

On se rend compte que la sensibilité calculée ne correspond pas tout à fait à celle donnée dans les caractéristiques. Il conviendra donc de faire attention à cette problématique lors de notre application et de vérifier les valeurs mesurées (voir chapitre sur la vérification des valeurs de la centrale).

6.3 Vérification des mesures calculées par la bibliothèque

Afin d'anticiper une éventuelle utilisation de la centrale sans utiliser la bibliothèque je vous propose de programmer vous-même le calcul des valeurs réelles à partir des valeurs brutes comme le fait la bibliothèque.

Afin de faciliter la modification ultérieure du programme pour l'adapter à une centrale inertielles différente, j'utilise la déclaration de **constante** avec l'instruction **#define**.

Ceci permet de n'avoir qu'une valeur à modifier facilement accessible en début de programme sans avoir à corriger un grand nombre d'instructions réparties dans le programme.

Le [deuxième programme](#) permet de calculer les valeurs et afficher en plus des précédentes pour vérifier si elles correspondent bien aux valeurs calculées par la bibliothèque.

```
#define sensi_acc      488
#define sensi_gyr      70000

#include "LSM6DS3.h"
#include "Wire.h"

LSM6DS3 myIMU(I2C_MODE, 0x6A);

int16_t ax_brut, ay_brut, az_brut, gx_brut, gy_brut, gz_brut;
float Ax,Ay,Az,Gx,Gy,Gz;

void setup() {
  Serial.begin(115200);
  if (myIMU.begin() != 0) {
    Serial.println("Problème de connexion avec la centrale");
  } else {
    Serial.println("Centrale prête à fonctionner ! ");
  }
}

void loop() {
  ax_brut = myIMU.readRawAccelX();
  ay_brut = myIMU.readRawAccelY();
  az_brut = myIMU.readRawAccelZ();
  gx_brut = myIMU.readRawGyroX();
  gy_brut = myIMU.readRawGyroY();
  gz_brut = myIMU.readRawGyroZ();

  float ax_biblio = myIMU.readFloatAccelX();
  float ay_biblio = myIMU.readFloatAccelY();
  float az_biblio = myIMU.readFloatAccelZ();
  float gx_biblio = myIMU.readFloatGyroX();
  float gy_biblio = myIMU.readFloatGyroY();
  float gz_biblio = myIMU.readFloatGyroZ();

  Ax = float(ax_brut) * sensi_acc / 1000000;
  Ay = float(ay_brut) * sensi_acc / 1000000;
  Az = float(az_brut) * sensi_acc / 1000000;
  Gx = float(gx_brut) * sensi_gyr / 1000000;
  Gy = float(gy_brut) * sensi_gyr / 1000000;
  Gz = float(gz_brut) * sensi_gyr / 1000000;

  Serial.println(" -- Accéléromètre : --");
  Serial.print(" - X_raw = "); Serial.print(ax_brut); Serial.print(" - X_réel = "); Serial.print(ax_biblio,2);
  Serial.print(" g"); Serial.print(" - X_calc = "); Serial.print(Ax,2); Serial.println(" g");
  Serial.print(" - Y_raw = "); Serial.print(ay_brut); Serial.print(" - Y_réel = "); Serial.print(ay_biblio,2);
  Serial.print(" g"); Serial.print(" - Y_calc = "); Serial.print(Ay,2); Serial.println(" g");
  Serial.print(" - Z_raw = "); Serial.print(az_brut); Serial.print(" - Z_réel = "); Serial.print(az_biblio,2);
  Serial.print(" g"); Serial.print(" - Z_calc = "); Serial.print(Az,2); Serial.println(" g");
  Serial.println(" -- Gyromètre : --");
  Serial.print(" - X_raw = "); Serial.print(gx_brut); Serial.print(" - X_réel = "); Serial.print(gx_biblio,2);
  Serial.print(" °/s"); Serial.print(" - X_calc = "); Serial.print(Gx,2); Serial.println(" °/s");
  Serial.print(" - Y_raw = "); Serial.print(gy_brut); Serial.print(" - Y_réel = "); Serial.print(gy_biblio,2);
  Serial.print(" °/s"); Serial.print(" - Y_calc = "); Serial.print(Gy,2); Serial.println(" °/s");
  Serial.print(" - Z_raw = "); Serial.print(gz_brut); Serial.print(" - Z_réel = "); Serial.print(gz_biblio,2);
  Serial.print(" °/s"); Serial.print(" - Z_calc = "); Serial.print(Gz,2); Serial.println(" °/s");
  Serial.println();
  Serial.println();
  delay(500);
}
```

Remarques : Pour la suite des programmes, les valeurs réelles seront donc celles calculées directement.

6.4 Nécessité d'un étalonnage

On vient de voir que les valeurs des vitesses angulaires ne sont pas nécessairement nulles lorsque le capteur est parfaitement immobile.

De plus lorsqu'une centrale inertuelle (carte électronique) est « installée » dans un système, il n'est pas certain que celle-ci soit toujours **parfaitement horizontale en position initiale**.

Cette contrainte implique que les valeurs d'accélération sur les axes x et y ne seront alors pas nulles et la composante sur z ne sera pas parfaitement égale à 1g.

Une méthode classique dans l'utilisation des capteurs consiste en un **étalonnage** pour définir des **offsets** qu'il faudra retirer aux valeurs brutes pour calculer la valeur réelle :

$$\text{Valeur réelle} = (\text{Valeur brute} - \text{Offset}) \times \text{sensibilité}$$

Le code fourni ci-dessous propose une solution simple qui consiste à calculer une **valeur moyenne** sur un **nombre d'échantillons** qu'il conviendra de lancer au démarrage de la centrale.

```
//Variables pour la lecture des offsets
int32_t ax_offset, ay_offset, az_offset;
int32_t gx_offset, gy_offset, gz_offset;

void etalonnage()
{
    //Variable locale pour définir le nombre d'itération du calcul
    int nombre_echantillons = 1000;
    //Réinitialisation des offsets
    ax_offset = 0;
    ay_offset = 0;
    az_offset = 0;
    gx_offset = 0;
    gy_offset = 0;
    gz_offset = 0;
    //Boucle de sommation de chaque valeur pour le calcul de la moyenne
    for (int i = 1; i <= nombre_echantillons; i++) {
        lecture_valeurs_brutes();
        ax_offset += ax_brut;
        ay_offset += ay_brut;
        az_offset += az_brut;
        gx_offset += gx_brut;
        gy_offset += gy_brut;
        gz_offset += gz_brut;
    }
    //Calcul de la moyenne des valeurs mesurées pendant les itérations
    ax_offset /= nombre_iterations;
    ay_offset /= nombre_iterations;
    az_offset /= nombre_iterations;
    gx_offset /= nombre_iterations;
    gy_offset /= nombre_iterations;
    gz_offset /= nombre_iterations;
    //Rectification de l'offset sur Z pour qu'il donne 1g en statique (fonction de la sensibilité)
    az_offset -= ####;
}
```

A FAIRE: Ajouter cette routine à votre code, en spécifiant le calcul de rectification de l'offset de l'accélération sur Z pour que le capteur indique bien 1g en statique.

A FAIRE: Ajouter l'appel de cette fonction dans le setup pour qu'elle se lance, **une seule fois**, au démarrage de l'Arduino et tester votre code.

7 ESTIMATION DE L'ATTITUDE

Outre les mesures d'accélération et de vitesse angulaire instantanées, une des utilisations courantes des centrales inertielles concerne **l'estimation de l'attitude d'un solide dans l'espace**.

On parle de capteur AHRS « Attitude and Heading Reference System » qui intègrent aussi un magnétomètre en plus de l'accélémètre et du gyromètre.

Il s'agit de pouvoir définir précisément la position et l'inclinaison de la centrale dans les 3 dimensions et généralement par rapport à un référentiel fixe.

7.1 Le problème de Wahba

Posé par Grace Wahba en 1965, il consiste à définir l'**orientation** d'un objet par rapport à une **base de référence** et répondre à la question suivante :

A partir de deux vecteurs non colinéaires et non nuls, mesurés dans le système de coordonnées mobiles et, déjà connus dans le système de coordonnées fixes, est-il possible de déterminer l'attitude ?

Les vecteurs connus correspondent au champ de gravité et éventuellement au champ magnétique terrestre.

Le problème de Wahba consiste donc à utiliser les redondances entre les données ainsi que l'incertitude de chaque mesure pour obtenir la meilleure estimation possible de l'orientation de l'objet.

Ce problème a fait l'objet de nombreuses recherches et publication et le [chapitre fusion de données](#) présente certaines méthodes et algorithmes permettant de satisfaire au mieux cette estimation.

Le problème de Wahba et quelques algorithmes de résolution A. Garcia, F. Laurent, F-X. Morel March 30, 2010

L'estimation de l'attitude correspondra ici à l'estimation des angles d'inclinaison de la centrale, par rapport à une **position initiale et par rapport à un référentiel fixe.**

7.2 Les différentes méthodes pour l'orientation d'un solide dans l'espace

Les théories sur la mesure d'attitude à partir d'une centrale inertuelle, proposent plusieurs solutions pour représenter l'orientation d'un solide dans l'espace dont les principales sont :

- ✓ Les Angles d'Euler classiques
- ✓ **Les Angles de LACET, ROULIS et TANGAGE** : solution retenue dans ce document.
- ✓ Les matrice de rotation et cosinus directeurs
- ✓ La représentation angle/axe
- ✓ Les paramètres d'Euler complets aussi appelés **QUATERNIONS UNITAIRES**

Habituellement, les changements de repères en 3D, utilisent des **matrices de rotations**, appelées aussi matrices de passage ou matrices de changement de base.

L'outil Mathématiques « matrice » n'étant pas au programme de Sciences de l'ingénieur ou STI2D en pré bac, les matrices de rotations ne sont, volontairement, pas utilisées ici.

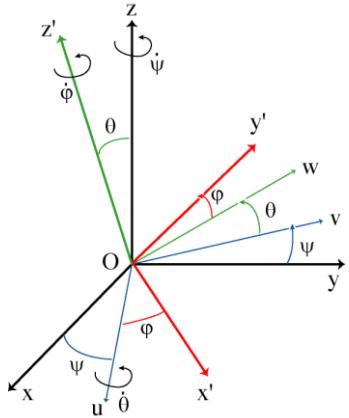
Les démonstrations proposées pour les calculs présentés en annexes, sont donc limitées à l'utilisation d'expressions algébriques « simples » qui pourront être comprises par un élève de Terminale.

Les quaternions, ne seront pas développés ici mais ils présentent des avantages et sont très utilisés pour leur « simplicité » de notation. Ils sont présentés succinctement avec le [problème du blocage de cardan](#).

7.3 Paramétrage de l'orientation avec les angles d'EULER

Définis au 18^{ème} siècle, les trois angles d'Euler sont utilisés en théorie pour orienter un référentiel $(\mathbf{O}, \vec{x}', \vec{y}', \vec{z}')$ lié à un solide par rapport à un référentiel fixe $(\mathbf{O}, \vec{x}, \vec{y}, \vec{z})$ par trois rotations successives :

- ✓ La **précession**, d'angle ψ autour de l'axe (\mathbf{O}, \vec{z}) qui fait passer de $(\mathbf{O}, \vec{x}, \vec{y}, \vec{z})$ au référentiel $(\mathbf{O}, \vec{u}, \vec{v}, \vec{z})$.
- ✓ La **nutation**, d'angle θ autour de l'axe (\mathbf{O}, \vec{u}) (ou ligne des nœuds), qui fait passer de $(\mathbf{O}, \vec{u}, \vec{v}, \vec{z})$ à $(\mathbf{O}, \vec{u}, \vec{w}, \vec{z}')$.
- ✓ La **rotation propre**, ou giration, d'angle ϕ autour de l'axe (\mathbf{O}, \vec{z}') , qui fait passer de $(\mathbf{O}, \vec{u}, \vec{w}, \vec{z}')$ au référentiel lié au solide $(\mathbf{O}, \vec{x}', \vec{y}', \vec{z}')$.



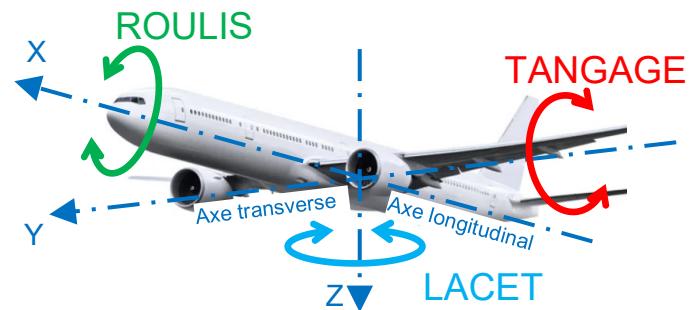
[Les Angles d'Euler sur Wikipédia](#)

[Marc Renaud. Comment définir l'orientation d'un corps ? Rapport LAAS n° 96078. 2017. fhal01523925](#)

7.4 Paramétrage de l'orientation avec les angles de « CARDAN »

En mécanique, et en particulier en navigation maritime ou aéronautique, l'orientation d'un solide dans l'espace utilise une définition plus récente du paramétrage de l'orientation avec 3 angles dit de « cardan » :

- ✓ la rotation de **LACET (Yaw)** d'angle ψ : le **CAP** (*angle entre l'axe longitudinal du solide et le pôle magnétique ou géographique*) ;
- ✓ la rotation de **TANGAGE (Pitch)** d'angle θ : l'**ASSIETTE** (*inclinaison autour de l'axe longitudinal : montée / descente*) ;
- ✓ la rotation de **ROULIS (Roll)** d'angle ϕ : la **GITE** sur un bateau (*inclinaison autour de l'axe longitudinal : droite / gauche*).

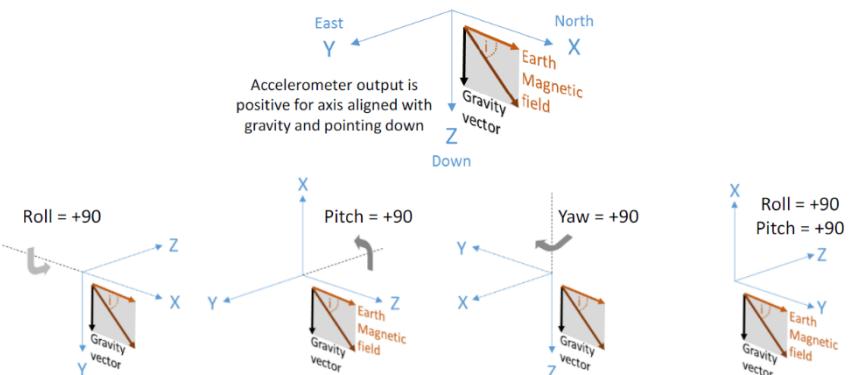


La bibliographie parle parfois d'angles nautiques ou angles de Tait-Bryan mais il est difficile de remonter à la source historique de cette définition.

En aéronautique la [convention NED](#) utilise une orientation de l'axe Z vers le bas pour obtenir un tangage positif lors de la montée.

On retrouve cette définition dans certaines documentations des centrales inertielles.

Ce choix n'a pas été retenu ici et toutes les illustrations conservent un axe Z vertical vers le haut afin de correspondre aux axes définis, par défaut, dans notre centrale inertuelle.



Roll, Pitch, Yaw degrees	Acc (Gravity vector) $G = 1g = 1000mg$	Mag (Earth mag field) $B = 10.90\text{mT} = 100.900\text{ mG}$
0, 0, 0	0, 0, +G	+B Cos(i), 0, +B Sin(i)
+90, 0, 0	0, +G, 0	+B Cos(i), +B Sin(i), 0
0, +90, 0	-G, 0, 0	-B Sin(i), 0, +B Cos(i)
0, 0, +90	0, 0, +G	0, -B Cos(i), +B Sin(i)
+90, +90, 0	-G, 0, 0	-B Sin(i), +B Cos(i), 0

7.5 Paramétrage des angles de ROULIS et TANGAGE

L'estimation de la mesure du cap par l'angle de LACET se doit d'être **absolue** par rapport au NORD terrestre ou magnétique.

- ✓ Avec un accéléromètre il ne sera pas possible de calculer un angle de LACET (voir figure ci-contre).

« Dans cette figure, nous avons dessiné un parallélépipède en vert, les axes x , y et z et le vecteur g d'accélération de pesanteur (diagonale du parallélépipède).

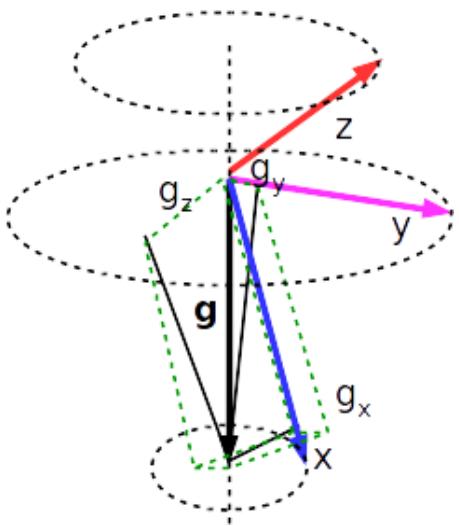
Imaginez cet ensemble comme un solide et faites-le tourner autour de l'axe vertical d'un angle quelconque.

Alors aucune composante de g ne change mais le repère (x,y,z) n'est plus le même puisqu'il a tourné.

Ainsi la connaissance de g_x , g_y , g_z ne suffira pas pour trancher parmi tous les repères possibles, lequel est celui que l'on cherche.

Votre accéléromètre est insensible aux rotations par rapport à un axe vertical et sera donc insuffisant pour retrouver l'attitude. »

Micro contrôleur AVR/Travail pratique/Utilisation d'un Accéléromètre MPU6050 - fr.wikiversity.org



[Illustration fr.wikiversity.org](http://fr.wikiversity.org)

- ✓ Avec le gyromètre, même s'il est possible de calculer une rotation autour de l'axe Z , elle ne sera que **relative** par rapport à la **position initiale**.
- ✓ Pour obtenir une information fiable de **LACET** il est donc nécessaire d'utiliser un autre capteur de type **compas** ou **magnétomètre** mais n'est pas l'objet d'étude de ce document.

Pour cette étude nous nous limiterons donc à calculer les angles de **ROULIS** et de **TANGAGE** à partir des données d'accélération et de vitesses angulaires.

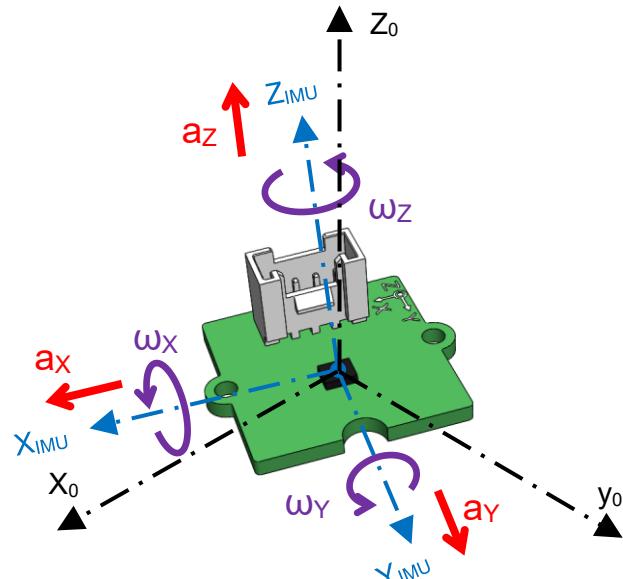
Il sera tout de même possible d'obtenir une information de LACET « **RELATIVE** » avec le gyromètre.

On considère donc un référentiel $(\mathbf{0}, \overrightarrow{\mathbf{X}_0}, \overrightarrow{\mathbf{Y}_0}, \overrightarrow{\mathbf{Z}_0})$ qui correspondrait au repère $(\mathbf{0}, \vec{\mathbf{u}}, \vec{\mathbf{v}}, \vec{\mathbf{z}})$ des angles d'Euler après l'orientation en lacet.

Pour le choix de l'axe longitudinal il est possible d'utiliser l'axe $\overrightarrow{\mathbf{X}_{IMU}}$ ou l'axe $\overrightarrow{\mathbf{Y}_{IMU}}$, ce premier choix sera fonction de l'implantation de la puce électronique ou de l'implantation du capteur sur le système à mesurer.

L'étude qui suit impose le choix de l'axe $\overrightarrow{\mathbf{X}_{IMU}}$ comme axe longitudinal.

Tous les calculs se feront autour d'une **position initiale de repos** qui sera considérée « **parfaitement** » horizontale avec l'axe $\overrightarrow{\mathbf{Z}_{IMU}}$ aligné avec la verticale $\overrightarrow{\mathbf{Z}_0}$.

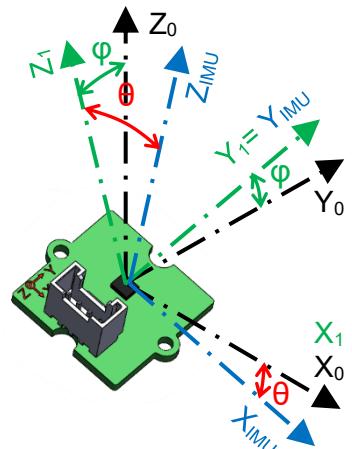


En réalité la position initiale ne sera pas toujours parfaitement horizontale et il sera, la plupart du temps, nécessaire de réaliser un **étalonnage** pour calculer des **OFFSETS** qu'il conviendra de soustraire aux données brutes pour obtenir les valeurs réelles.

Le paramétrage de l'attitude de notre capteur se limitera donc à **2 rotations** qui peuvent se définir de 3 manières différentes :

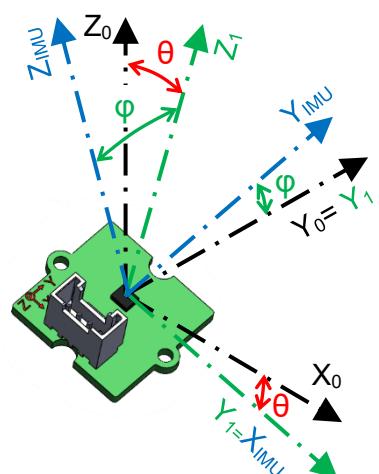
7.5.1 CHOIX N°1 : Première rotation de ROULIS puis celle de TANGAGE

- ✓ La rotation de **ROULIS** d'un angle φ autour de l'axe \vec{X}_0 auquel on associe un repère $(\mathbf{0}, \vec{X}_1, \vec{Y}_1, \vec{Z}_1)$
- ✓ La rotation de **TANGAGE** d'un angle θ autour de l'axe \vec{Y}_1 qui permet de définir le repère de la centrale $(\mathbf{0}, \vec{X}_{\text{IMU}}, \vec{Y}_{\text{IMU}}, \vec{Z}_{\text{IMU}})$



7.5.2 CHOIX N°2 : Première rotation de TANGAGE puis celle de ROULIS

- ✓ La rotation de **TANGAGE** d'un angle θ autour de l'axe \vec{Y}_0 auquel on associe un repère $(\mathbf{0}, \vec{X}_1, \vec{Y}_1, \vec{Z}_1)$
- ✓ La rotation de **ROULIS** d'un angle φ autour de l'axe \vec{X}_1 qui permet de définir le repère de la centrale $(\mathbf{0}, \vec{X}_{\text{IMU}}, \vec{Y}_{\text{IMU}}, \vec{Z}_{\text{IMU}})$



7.5.3 CHOIX N°3 : Les 2 angles indépendants

- ✓ La rotation de **ROULIS** d'un angle φ autour de l'axe \vec{X}_0
- ✓ La rotation de **TANGAGE** d'un angle θ autour de l'axe \vec{Y}_0

Cette solution ne découle pas d'une définition géométrique classique de changement de repères. Mais on la retrouve parfois dans certaines documentations techniques fournies par certains constructeurs de matériel.

8 CALCUL DES ANGLES AVEC L'ACCELEROMETRE

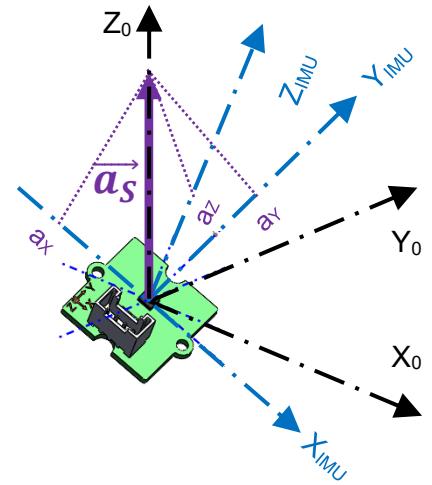
L'accéléromètre fournit les 3 composantes de l'accélération suivant les 3 axes du capteur dans le repère $(\mathbf{0}, \overrightarrow{\mathbf{X}_{IMU}}, \overrightarrow{\mathbf{Y}_{IMU}}, \overrightarrow{\mathbf{Z}_{IMU}})$:

$$\overrightarrow{a_{IMU}} = \begin{vmatrix} a_X \\ a_Y \\ a_Z \end{vmatrix} \quad \text{ou} \quad \overrightarrow{a_{IMU}} = a_X \cdot \overrightarrow{\mathbf{X}_{IMU}} + a_Y \cdot \overrightarrow{\mathbf{Y}_{IMU}} + a_Z \cdot \overrightarrow{\mathbf{Z}_{IMU}}$$

EN STATIQUE, et de par son architecture, le capteur mesure une accélération \vec{a}_S , image de l'accélération gravitationnelle donc verticale mais de composante *positive* (*voir le problème de l'accélération verticale*).

La mesure de l'accélération de pesanteur peut se définir dans le repère $(\mathbf{0}, \overrightarrow{\mathbf{X}_0}, \overrightarrow{\mathbf{Y}_0}, \overrightarrow{\mathbf{Z}_0})$:

$$\overrightarrow{a_S} = \begin{vmatrix} 0 \\ 0 \\ +\|\vec{g}\| \end{vmatrix} \quad \text{ou} \quad \overrightarrow{a_S} = \|\vec{g}\| \cdot \overrightarrow{\mathbf{Z}_0}$$



Lorsque le capteur est parfaitement horizontal, seule la composante a_z mesure **l'opposée de la gravité** ($-g$), les composantes a_x et a_y sont alors nulles.

Lorsque le capteur s'incline il est alors possible de calculer les angles de roulis et tangage grâce aux variations des composantes lues par le capteur.

Les calculs des angles diffèrent en fonction du [choix du paramétrage des angles](#). Les démonstrations des calculs pour obtenir ces formules sont [détailles en annexe](#).

8.1 CHOIX N°1 : ROULIS puis TANGAGE

ROULIS

$$\varphi = \text{ArcTAN} \left(\frac{a_y}{\pm \sqrt{a_x^2 + a_z^2}} \right)$$

TANGAGE

$$\theta = \text{ArcTAN} \left(-\frac{a_x}{a_z} \right)$$

8.2 CHOIX N°2 : TANGAGE puis ROULIS

ROULIS

$$\varphi = \text{ArcTAN} \left(\frac{a_y}{a_z} \right)$$

TANGAGE

$$\theta = \text{ArcTAN} \left(\frac{a_x}{\pm \sqrt{a_y^2 + a_z^2}} \right)$$

8.3 CHOIX N°3 : TANGAGE et ROULIS indépendants

ROULIS

$$\varphi = \text{ArcTAN} \left(\frac{a_y}{\pm \sqrt{a_x^2 + a_z^2}} \right)$$

TANGAGE

$$\theta = \text{ArcTAN} \left(\frac{a_x}{\pm \sqrt{a_y^2 + a_z^2}} \right)$$

8.4 La fonction ATAN2 pour réduire les limites de l'arc tangente

La fonction trigonométrique Arctan possède plusieurs inconvénients :

- ✓ Elle n'est valide que dans l'intervalle $]-90^\circ; 90^\circ[$;
- ✓ la valeur particulière de l'ArcTAN quand $x = 0$ renvoie une erreur de calcul puisque la fonction tan n'est pas définie ;
- ✓ Le problème du **blocage de cardan** (gimbal lock voir [chapitre associé](#)) qui implique des singularités (indéterminations) pour certaines positions particulières.

Afin de limiter certaines de ces problématiques, il est possible d'utiliser la fonction numérique **Atan2**, (voir [ANNEXE : La fonction numérique ATAN2](#)) qui permet de calculer l'angle dans l'intervalle $]-180^\circ; 180^\circ[$;

La correspondance avec l'Arctan est directe exemple pour le choix N°1 :

$$\varphi = \text{ArcTAN} \left(\frac{a_y}{\pm \sqrt{a_x^2 + a_z^2}} \right) \iff$$

ROULIS

$$\varphi = \text{atan2}(a_y, \pm \sqrt{a_x^2 + a_z^2})$$

$$\theta = \text{ArcTAN} \left(-\frac{a_x}{a_z} \right) \iff$$

TANGAGE

$$\theta = \text{atan2}(-a_x, a_z)$$

Le calcul du roulis nécessite donc de connaître le sens du vecteur accélération, le plus simple est d'utiliser Un test conditionnel de la composante de l'accélération sur Z :

```
if ( az > 0 )
{
    Calcul avec la composante positive
}
else
{
    Calcul avec la composante négative
}
```

La fonction racine carrée se code en utilisant sqrt(valeur) et la fonction carrée avec pow(valeur , puissance).

La fonction numérique atan2(opposé , adjacent) est directement utilisable dans l'IDE ([voir annexe](#)) mais elle renvoie une valeur en radian.

Pour convertir les radians en degrés la bibliothèque Arduino.h intègre directement la constante de PI mais aussi la constante RAD_TO_DEG qu'il est possible d'utiliser nativement.

```
#define PI 3.1415926535897932384626433832795
#define HALF_PI 1.5707963267948966192313216916398
#define TWO_PI 6.283185307179586476925286766559
#define DEG_TO_RAD 0.017453292519943295769236907684886
#define RAD_TO_DEG 57.295779513082320876798154814105
```

8.5 Programmation du calcul des angles avec l'acceleromètre

Pour cette application on vous propose d'utiliser l'axe $\overrightarrow{X_{IMU}}$ comme axe longitudinal ainsi que le choix N°1 du paramétrage de l'attitude du capteur, ROULIS puis TANGAGE :

- ✓ La rotation de **ROULIS** d'un angle φ autour de l'axe $\overrightarrow{X_0}$ auquel on associe un repère $(\mathbf{0}, \overrightarrow{X_1}, \overrightarrow{Y_1}, \overrightarrow{Z_1})$
- ✓ La rotation de **TANGAGE** d'un angle θ autour de l'axe $\overrightarrow{Y_1}$ qui permet de définir le repère de la centrale $(\mathbf{0}, \overrightarrow{X_{IMU}}, \overrightarrow{Y_{IMU}}, \overrightarrow{Z_{IMU}})$.

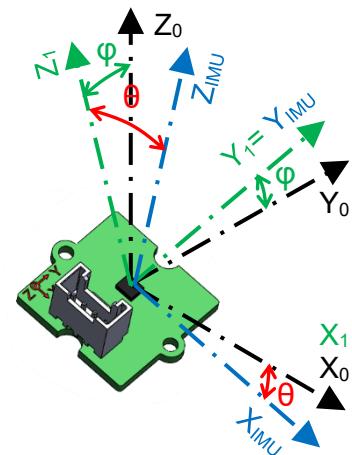
8.5.1 Intégration des calculs au programme

Le code proposé dans le programme fournit ajoute aux calculs précédents les variables et les le calcul :

```
float roulis_acc, tangage_acc;

void loop()
{
    //Calculs des angles à partir de l'accéléromètre
    roulis_acc = ;
    tangage_acc = ;

    Serial.print(" | ROULIS Acc = "); Serial.print(roulis_acc, 0); Serial.print("°");
    Serial.print(" | TANGAGE Acc = "); Serial.print(tangage_acc, 0);
    Serial.print("°");
}
```



A FAIRE: Dans le programma ajouter les codes de calculs des 2 angles.

A FAIRE: Transférer votre programme et ouvrir le moniteur série.

8.5.1 Vérifier les valeurs calculées

A FAIRE: Vérifier que les valeurs affichées soient cohérentes (vous pouvez compléter le tableau page suivante).

Vérification des valeurs calculées par l'accéléromètre.

Positions			Ax	Ay	Az	ROULIS	TANGAGE
Rotation autour de l'axe Y positive	+90°						
	+45°						
Rotation autour de l'axe X positive	+90°						
	+60°						
	+45°						
POSITION INITIALE	0°						
Rotation autour de l'axe X négative	-45°						
	-60°						
	-90°						
Rotation autour de l'axe Y positive	-45°						
	-90°						

9 CALCUL DES ANGLES AVEC LE GYROMETRE

9.1 De la vitesse à l'angle

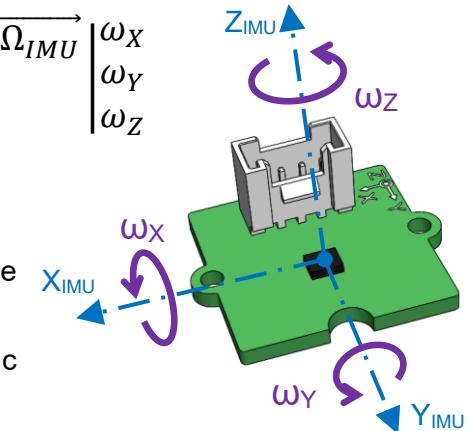
Le gyromètre mesure le vecteur vitesse angulaire instantané $\vec{\Omega}_{IMU}$, dans le repère de la centrale inertie (O, \vec{X}_{IMU} , \vec{Y}_{IMU} , \vec{Z}_{IMU}) :

Il fournit 3 valeurs en **deg/s**, correspondantes aux composantes du **vecteur vitesse angulaire** du capteur, autour de chacun des 3 axes.

Par définition, la vitesse instantanée ω , notée aussi θ' , est la dérivée de la variation angulaire θ sur un intervalle de temps.

Connaissant la vitesse de rotation entre 2 instants t_0 et t_1 , il est donc possible de calculer la variation angulaire $\theta_{0 \rightarrow 1}$ par intégration :

$$\omega = \frac{d\theta}{dt} = \theta' \rightarrow \theta_{0 \rightarrow 1} = \int_{t_0}^{t_1} \theta' \cdot dt$$



9.2 Nécessité d'une intégration numérique

Pour un **calcul numérique** avec un microcontrôleur, le calcul intégral peut être simplifié par une **sommation** des variations angulaires au cours du temps :

$$\theta = \sum \theta' \cdot \Delta t \quad \theta_{\text{total}} = \theta_{\text{ précédent}} + \theta' \cdot \Delta t$$

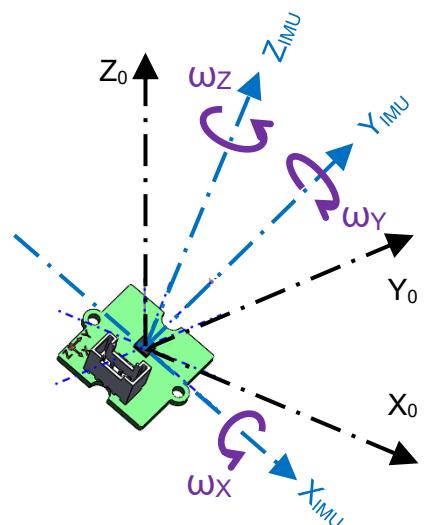
Le problème consistera à créer un calcul à **INTERVALLE REGULIER**.

9.3 Calcul numérique des angles à partir de la vitesse angulaire

Contrairement à l'accéléromètre le gyromètre est capable de calculer un angle de **LACET** mais il sera **relatif** par rapport à la position initiale.

Le calcul de chaque angle est donc parfaitement indépendant et peut se faire à partir des 3 formules suivantes :

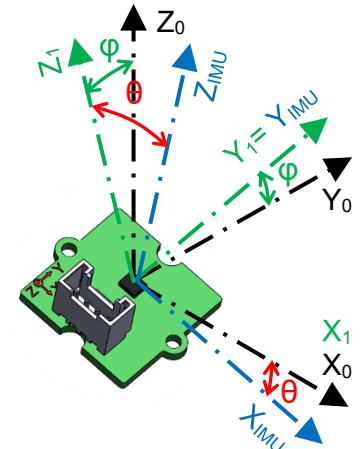
- ✓ ROULIS : $\phi_{\text{total}} = \phi_{\text{ précédent}} + \omega_X \cdot \Delta t$
- ✓ TANGAGE : $\theta_{\text{total}} = \theta_{\text{ précédent}} + \omega_Y \cdot \Delta t$
- ✓ LACET : $\psi_{\text{total}} = \psi_{\text{ précédent}} + \omega_Z \cdot \Delta t$



9.4 Programmation du calcul des angles avec le gyromètre

Comme pour l'exemple précédent, on conserve les mêmes choix : \vec{X}_{IMU} comme axe longitudinal ainsi que ROULIS puis TANGAGE :

- ✓ La rotation de ROULIS d'un angle φ autour de l'axe \vec{X}_0 auquel on associe un repère $(\mathbf{0}, \vec{X}_1, \vec{Y}_1, \vec{Z}_1)$
- ✓ La rotation de TANGAGE d'un angle θ autour de l'axe \vec{Y}_1 qui permet de définir le repère de la centrale $(\mathbf{0}, \vec{X}_{\text{IMU}}, \vec{Y}_{\text{IMU}}, \vec{Z}_{\text{IMU}})$.



9.4.1 Comment calculer à intervalle régulier ?

Les calculs d'intégration numériques nécessitent de définir une période $\Delta t = (t_1 - t_0)$ relativement petite pour que la vitesse n'ait pas le temps de varier trop fortement mais surtout relativement précise.

Pour gérer le temps, les microcontrôleurs possèdent des fonctions `millis()` ou `micros()` que l'on peut utiliser facilement.

Un pseudo code est présenté ci-dessous pour expliquer le calcul à intervalle régulier :

1. Enregistrer le temps initial (dans le setup) exemple : $t_0 = \text{millis}()$
2. Dans la boucle principale (loop) :
 - ✓ Calculer la période $\Delta t = \text{millis}() - t_0$
 - ✓ Procéder au calcul numérique si la période Δt est supérieure à une valeur T_{ech} choisie :

Si $\Delta t \geq T_{\text{ech}}$:

$$\theta_{\text{total}} = \theta_{\text{précédent}} + \theta' \cdot \Delta t$$

$$t_0 = \text{millis}()$$
 (en pensant à réinitialiser le temps initial)

Plutôt que de définir une période, on vous propose ici de définir une fréquence d'échantillonnage.

On utilise pour cela une constante `f_ech` en Hz, personnalisable au début du programme avec la fonction `#define`.

La variable locale `T_ech` utilisée pour le calcul de la période n'est pas indispensable mais elle sert aussi aux calculs des angles par intégration.

Le code proposé dans le programme fournit reprend l'ensemble des calculs précédents en ajoutant les variables et calcul à intervalle régulier ainsi que les codes d'affichage des valeurs sur le moniteur série :

```
#define      f_ech 100

float      roulis_gyr, tangage_gyr, lacet_gyr;
uint32_t    t0;

void setup()
{
    t0 = millis();
}

void loop()
{
    uint16_t T_ech = millis() - t0;

    if ( T_ech >= ( 1000 / f_ech ) )
    {
        roulis_gyr += Gx * float(T_ech) / 1000;
        tangage_gyr += Gy * float(T_ech) / 1000;
        lacet_gyr   += Gz * float(T_ech) / 1000;

        t0 = millis();
    }
}
```

A FAIRE: Ouvrir le fichier exemple, transférer le programme dans l'arduino et tester le code.

9.4.2 Vérification des angles calculés

A FAIRE: Tester le code afin de vérifier que les angles calculés soient cohérents par rapport aux angles de l'accéléromètre.

A FAIRE: Procéder à un test en ne faisant varier qu'un seul des 2 angles.

A FAIRE: Réfléchir aux causes probables si vous rencontrez une différence notable.

9.5 Nécessité de prendre en compte l'inclinaison pour le calcul

Les angles doivent donc se calculer à partir des vitesses angulaires autour de leurs axes propres.

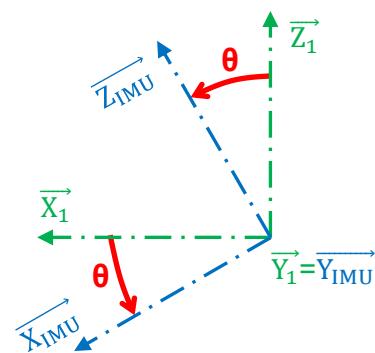
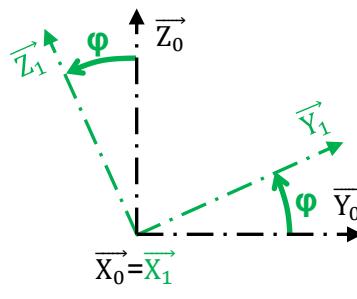
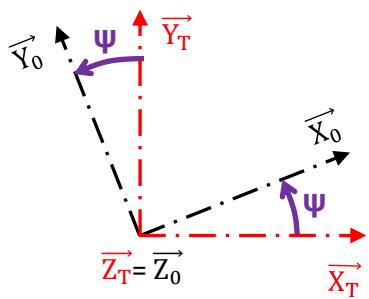
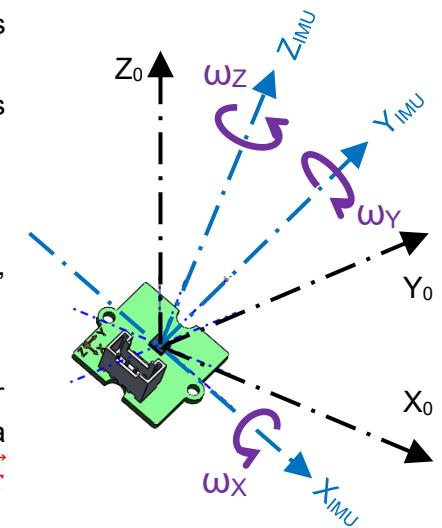
Mais les composantes du gyromètre, ω_X , ω_Y et ω_Z , sont calculées dans le repère de la centrale inertielle ($0, \vec{X}_{IMU}, \vec{Y}_{IMU}, \vec{Z}_{IMU}$).

9.5.1 Impact du paramétrage sur les calculs

Par rapport au paramétrage du roulis et du tangage il est possible, ici, de calculer le LACET et donc de tenir compte de la rotation supplémentaire.

Ce paramètre supplémentaire implique donc 12 possibilités différentes.

On se limitera pour cette étude au paramétrage du choix N°1 pour l'accéléromètre auquel on ajoute un angle supplémentaire ψ pour définir la base ($0, \vec{X}_0, \vec{Y}_0, \vec{Z}_0$) par rapport à une base Terrestre ($0, \vec{X}_T, \vec{Y}_T, \vec{Z}_T$) dont \vec{X}_T désignera le nord magnétique ou terrestre.



Le paramétrage sera donc le suivant dans l'ordre :

- ✓ rotation de LACET ψ autour de l'axe \vec{Z}_0 ,
- ✓ rotation de Le ROULIS φ autour de l'axe \vec{X}_0 ,
- ✓ rotation de TANGAGE θ autour de l'axe \vec{Y}_1 .

9.5.2 Calcul des angles à partir de la vitesse angulaire

Le calcul de chaque angle devra donc se faire autour de chacun de leur axe et, de ce fait, les vitesses angulaires à intégrer devront être celles autour de leurs axes respectifs :

- ✓ ROULIS : $\varphi_{total} = \varphi_{ précédent } + \varphi'. \Delta t$
- ✓ TANGAGE : $\theta_{total} = \theta_{ précédent } + \theta'. \Delta t$
- ✓ LACET : $\psi_{total} = \psi_{ précédent } + \psi'. \Delta t$

Avec φ' , θ' et ψ' , les vitesses angulaires de ROULIS, TANGAGE et LACET autour de chaque axe.

Remarque : En programmation C il existe la instruction `+=` qui simplifie le code : $\varphi += \varphi'. \Delta t$

9.5.3 L'impact de l'inclinaison sur le calcul

φ' , θ' et ψ' , sont donc les vitesses angulaires respectives de ROULIS, TANGAGE et LACET qu'il convient de calculer autour de leurs axes respectifs.

Les angles d'inclinaison de la centrale induisent donc des variations de ces composantes dont il faut tenir compte pour les calculs :

- ✓ Vitesse de ROULIS autour de l'axe \vec{X}_0 :

$$\varphi' = \omega_X \cdot \cos \theta + \omega_Z \cdot \sin \theta$$

- ✓ Vitesse de TANGAGE autour de l'axe \vec{Y}_1 :

$$\theta' = \omega_Y + (\omega_X \cdot \sin \theta - \omega_Z \cdot \cos \theta) \cdot \tan \varphi$$

- ✓ Vitesse de LACET autour de l'axe \vec{Z}_0 :

$$\psi' = \frac{-\omega_X \cdot \sin \theta + \omega_Z \cdot \cos \theta}{\cos \varphi}$$

Les démonstrations pour obtenir ces formules [sont données en annexe](#).

9.5.4 Un calcul plus complexe qu'il n'y paraît

En effet le calcul des angles φ , θ , et ψ se calculent donc avec leurs vitesses angulaires φ' , θ' et ψ' respectives mais ces dernières sont aussi impactées par les angles...

Le code proposé dans l'exemple fournit, utilise des variables **locales**, de type flottant, et calcule chaque vitesse φ' , θ' et ψ' en fonction des angles calculés à partir de l'**accéléromètre** :

```

float lacet_vitesse = (-Gx * sin(tangage_acc * DEG_TO_RAD) + Gz * cos(tangage_acc * DEG_TO_RAD)) / cos(roulis_acc * DEG_TO_RAD);
float roulis_vitesse = Gx * cos(tangage_acc * DEG_TO_RAD) + Gz * sin(tangage_acc * DEG_TO_RAD);
float tangage_vitesse = Gy + (-Gx * sin(tangage_acc * DEG_TO_RAD) - Gz * cos(tangage_acc * DEG_TO_RAD)) * tan(roulis_acc * DEG_TO_RAD);

uint16_t T_ech = millis() - t0;

if (T_ech >= (1000 / f_ech))
{
    lacet_gyr += lacet_vitesse * float(T_ech) / 1000;
    roulis_gyr += roulis_vitesse * float(T_ech) / 1000;
    tangage_gyr += tangage_vitesse * float(T_ech) / 1000;

    t0 = millis();
}

```

Les fonctions trigonométriques `sin()` et `cos()` utilisent des angles en radians il est donc nécessaire de convertir les angles avec la constante `DEG_TO_RAD`

Attention aux unités, la variable locale `T_ech` de la période est en ms (voire en `μs`) et la vitesse angulaire est donnée par le capteur en $^{\circ}/s$.

A FAIRE : Ouvrir le programme fournit et transférer le dans le `μC` pour procéder aux tests suivants

10 VALIDITE DES ANGLES CALCULES

10.1 Validité des angles calculés en STATIQUE

A FAIRE: Utiliser la cale pentée pour incliner la centrale et vérifier les calculs d'angles

10.2 Validité de l'angle de LACET

A FAIRE: Poser la centrale sur un support plat et la faire tourner autour de l'axe Z afin de vérifier que l'angle de LACET calculé soit cohérent.

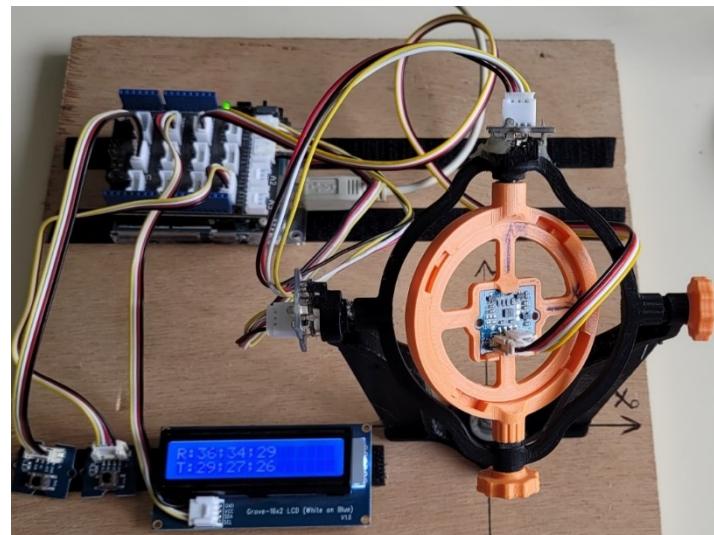
Il est possible de vérifier plus finement les angles à partir de dispositifs expérimentaux de type gyroscope.

Une platine prototype est proposée sur le [dépôt github IMU_Gyroscope](#).



C'est une sorte de gyroscope à 3 cardans dont 2 sont équipés de capteurs de rotations (potentiomètres) afin de proposer :

- ✓ la mesure la plus précise possible des angles « réels » entre cardans, correspondant au ROULIS et au TANGAGE,
- ✓ le calcul des angles à partir de la centrale,
- ✓ l'affichage sur un écran LCD des 6 angles (3 pour le ROULIS et 3 pour le TANGAGE) dans l'ordre suivant :
 - L'angle « réel » ;
 - L'angle calculé à partir de l'accéléromètre ;
 - L'angle calculé avec le gyromètre,
- ✓ la sortie, sur le moniteur série, de toutes les valeurs mesurées et calculées,
- ✓ la possibilité de bloquer le gyroscope dans une position fixe.



10.3 Nécessité d'une période de calcul précise.

L'utilisation des fonctions millis() et d'un test induit inévitablement une variation de la période, qu'il sera possible de « limiter » en lui préférant l'utilisation des **TIMER** et des **interruptions**, mais leur programmation nécessite des connaissances plus poussées.

Pour simplifier leur utilisation, il existe des bibliothèques comme [TimerOne](#) mais il faut prêter attention si des sorties PWM sont utilisées pour contrôler des MOTEURS ou des SERVOMOTEURS qui utilisent ces mêmes timers.

Mais même avec les Timers, ce type de calcul entraîne inévitablement une **dérive** qui viendra fausser le calcul.

10.4 Nécessité d'une modulation de l'angle

A FAIRE: Continuer à tourner de plus d'un $\frac{1}{2}$ tour. Est-ce que l'angle calculé par le gyromètre est cohérent ?

En effet il est facile de comprendre que la sommation de l'intégration numérique continuera au-delà des $\pm 180^\circ$ contrairement au calcul avec l'accéléromètre qui utilise la fonction atan2().

Il est donc nécessaire de moduler les valeurs d'angles calculées avec le gyromètre.

La fonction mathématique Modulo existe en C : `variable = dividende % diviseur` mais n'est valide que pour des valeurs entières.

Il existe une fonction `double fmod(double x, double y)` de la librairie `math.h` mais il serait nécessaire d'étudier la complexité temporelle de son algorithme afin de vérifier que le temps de traitement ne soit pas plus long que la période d'échantillonnage.

La solution simple proposée ici est d'utiliser une condition de limite de l'angle en cas de dépassement de la valeur 180° en positif ou négatif.

```
if (roulis_gyr < 180) roulis_gyr +=;
if (roulis_gyr >= 180) roulis_gyr -=;
if (tangage_gyr < 180) tangage_gyr +=;
if (tangage_gyr >= 180) tangage_gyr -=;
```

A FAIRE: Ajouter ces conditions après les calculs d'angles pour moduler les valeurs entre -180° et $+180^\circ$.

10.5 Le problème de la stabilité des mesures dans le temps

A FAIRE: Positionner votre centrale à plat en statique et observer l'évolution des valeurs des angles calculées avec l'accéléromètre et le gyromètre.

A FAIRE: Est-ce que les valeurs restent valides dans le temps ?

A FAIRE: Etudier les causes probables de ce phénomène et les éventuelles solutions afin de les limiter ?

10.6 Validité des angles calculés en DYNAMIQUE

A FAIRE: Maintenir la centrale à plat sur la table et déplacer la sur le plan horizontal d'avant en arrière sur les axes X puis Y et observer les valeurs calculées avec l'accéléromètre et le gyromètre.

A FAIRE: Est-ce que les valeurs restent valides en permanence ?

A FAIRE: Réfléchir aux causes probables et à une éventuelle solution afin de les limiter ?

11 LA FUSION DES CAPTEURS

11.1 Les problèmes de mesure

Nous avons donc vu comment obtenir des angles de LACET, ROULIS et TANGAGE à partir de l'accéléromètre et du gyromètre.

Mais les mesures présentent quelques problèmes inhérents :

- ✓ à l'architecture même du composant,
- ✓ au principe physique de la mesure,
- ✓ mais aussi aux calculs numériques.

11.1.1 Le problème du bruit de mesure

Le premier problème d'une mesure physique à partir d'un capteur analogique, vient du « **bruit** » correspondant à des signaux parasites perturbant la mesure.

Leurs sources peuvent être de multiples natures et entraînent un signal variable de faible amplitude mais de fréquence généralement assez élevée.

En effet, *même parfaitement STATIQUE et horizontal*, le signal envoyé par l'accéléromètre ne sera pas parfaitement nul sur l'axe x et y et variera autour de 1 pour la composante sur z.

Si l'amplitude du bruit dépasse un certain seuil, le capteur interprétera donc une valeur qui perturbe la mesure.

On peut retrouver les caractéristiques du bruit de notre centrale dans sa documentation technique.

Table 3. Mechanical characteristics

Symbol	Parameter	Test conditions	Min.	Typ. ⁽¹⁾	Max.	Unit
Rn	Rate noise density			7		mdps/ $\sqrt{\text{Hz}}$
An	Acceleration noise density	FS= $\pm 2 \text{ g}$ ODR = 104 Hz		90		$\mu\text{g}/\sqrt{\text{Hz}}$

Les centrales inertielles intègrent des FILTRES passe haut et passe bas que l'on peut configurer assez facilement. Voir chapitre sur [la configuration de la centrale](#).



11.1.2 Le problème de la mesure physique avec l'accéléromètre

Du fait de la mesure nécessairement STATIQUE, le calcul de l'angle avec l'accéléromètre ne sera possible que dans le cas où la centrale ne subit pas des variations de mouvements trop brusques.

En effet si le système rencontre des **accélérations ou décélérations importantes**, l'accéléromètre les mesure et les **angles calculés** seront alors **FAUSSES**.

11.1.3 Le problème du calcul numérique pour le gyromètre

Pour les valeurs issues du gyromètre, les mesures de vitesses angulaires ne seront pas trop perturbées par des accélérations brusques mais le calcul numérique, entraîne naturellement une **dérive** importante.

En effet même si le capteur est parfaitement au repos, les bruits de mesures et / ou les irrégularités de l'intervalle de calcul d'intégration **font varier la valeur de l'angle**.

11.2 La nécessité du filtrage ou de la fusion de données

La mesure de l'attitude d'un objet à partir de valeurs « imprécises » ou « bruitées », connue sous le nom de "problème de Wahba", a fait l'objet de nombreux travaux et publications de méthodes algorithmiques pour optimiser la réponse.

Parmi les plus célèbres, on trouve :

- ✓ la méthode TRIAD,
- ✓ la q-Method,
- ✓ l'algorithme QUEST,
- ✓ l'algorithme FQA (plutôt utilisées avec un accéléromètre et un magnétomètre).

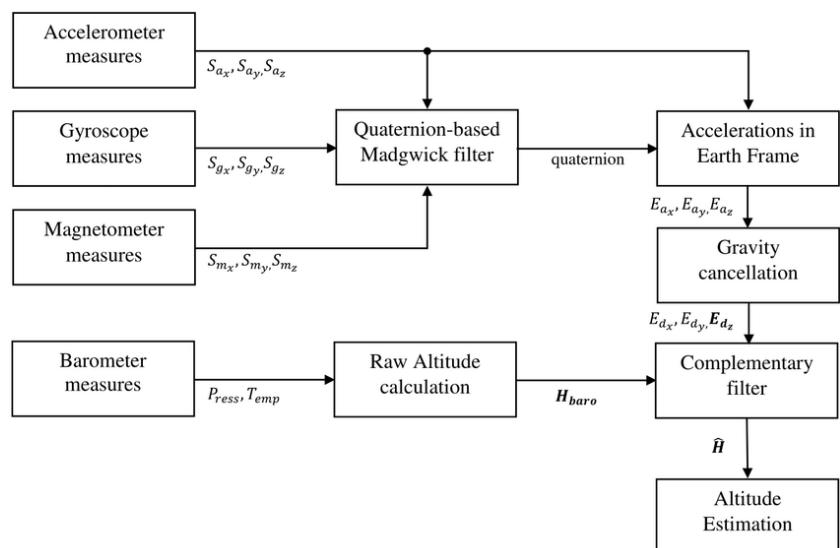
Tous ces algorithmes possèdent des degrés de complexités importants, utilisent des méthodes mathématiques comme celle des **moindres carrés** ou celle de la **racine de la moyenne des carrés** (RSMD) et sont difficilement compatibles avec une utilisation dans un microcontrôleur aux performances limitées.

La majorité de ces méthodes reposent sur la **FUSION DE DONNEES**, avec ou sans **FILTRAGE**, de plusieurs capteurs permettant d'optimiser le calcul d'attitude.

En réalité l'estimation de l'attitude d'un système repose sur plusieurs mesures qu'il faudra fusionner entre elles en utilisant plusieurs méthodes associées.

On peut représenter le principe sous forme de schéma bloc :

Pierleoni, Paola & Belli, Alberto & Maurizi, Lorenzo & Palma, Lorenzo & Pernini, Luca & Paniccia, Michele & Valenti, Simone. (2016). A Wearable Fall Detector for Elderly People Based on AHRS and Barometric Sensor. IEEE Sensors Journal. 16. 1-1. 10.1109/JSEN.2016.2585667.



[Illustration IEEE Sensors Journal](#)

Pour la suite on se contente de parler de filtrage à partir des données de l'accéléromètre et du gyromètre.

Le plus simple, souvent appelé dans la littérature **FILTRE COMPLEMENTAIRE**, est facile à implémenter.

Il correspond à un « Filtre à réponse impulsionnelle finie » (RIF) avec une fréquence de coupure identique en passe bas (pour l'accéléromètre) et passe haut (pour le gyromètre).

La plus utilisée dans la bibliographie est celle du **FILTRE DE KALMAN**, dont il existe plusieurs variantes, mais qui nécessite un traitement numérique assez « lourd ». Pour l'application avec un microcontrôleur simple, il est possible de tester une *version limitée* de ce filtre.

Plus récemment, Sébastien Madgwick a mis à disposition un travail de recherche [« An efficient orientation filter for inertial and inertial/magnetic sensor arrays »](#) que l'on retrouve sous le nom d'**ALGORITHME AHRS DE MADGWICK**.

Cet algorithme est proposé sous forme d'une [bibliothèque arduino](#) et pourra être implémenté « assez facilement ».

11.3 Le filtre COMPLEMENTAIRE

Ce « filtre simple » appelé "COMPLEMENTAIRE" fusionne les 2 angles avec un calcul « léger » :

$$\theta_{\text{fusionné}} = \alpha \cdot \theta_{\text{gyromètre}} + (1 - \alpha) \cdot \theta_{\text{accéléromètre}}$$

Il est souvent utilisé avec un coefficient α compris entre 96% et 99%.

En réalité le filtre complémentaire est issu d'une théorie développée depuis les années 1940 par Norbert Wiener, pour le filtrage du bruit d'un signal.

Il s'agit de filtrer les fréquences pour éliminer, ou en tous cas atténuer, les perturbations des signaux.

Ce calcul correspond à un **filtrage fréquentiel numérique** de type **RIF** ([Filtre à réponse impulsionnelle FINIE](#)) qui est limité, ici, au **premier ordre**.

Les développements des calculs à partir des transformés de Laplace ne seront pas évoqués ici.

Ce type de filtrage numérique échantillonné peut se calculer, à chaque échantillon **n**, par une équation d'ordre **N** avec des coefficients de gains **k_i**.

$$\theta_n = k_0 \cdot \theta_{\text{mesuré}} + k_1 \cdot \theta_{n-1} + k_2 \cdot \theta_{n-2} + \dots + k_N \cdot \theta_{n-N}$$

L'application de cette formule correspond donc à un filtre **RIF d'ordre 1** où ne sont prises en compte que les valeurs mesurées pour **l'échantillon n** sans tenir compte des précédents !

Le fait d'utiliser un coefficient **k = α** pour le **gyromètre** et **k = (1 - α)** pour l'**accéléromètre** correspond donc à un filtre de type **PASSE HAUT** pour le gyromètre et de type **PASSE BAS** pour l'accéléromètre dont les 2 fréquences de coupure sont identiques.

11.3.1 COEFFICIENT DE FILTRAGE ET FREQUENCE DE COUPURE

Il est possible de calculer le coefficient α , à partir de la période d'échantillonnage Δt et de la constante de temps τ du filtre :

$$\alpha = \frac{\tau}{\tau + \Delta t} \quad \Delta t = \frac{1}{f_{\text{ech}}} \quad \tau = \frac{1}{2\pi \cdot f_c}$$

Δt : période d'échantillonnage

τ : Constante de temps du filtre

f_c : fréquence de coupure

La fréquence de coupure doit, théoriquement, être inférieure à celle du bruit de l'accéléromètre.

Il est aussi possible de calculer le coefficient α à partir des fréquences : $\alpha = \frac{f_{\text{ech}}}{f_{\text{ech}} + 2\pi f_c}$

Exemples :

- ✓ Avec une fréquence d'échantillonnage $f_{\text{ech}} = 50 \text{ Hz}$ et une fréquence de coupure $f_c = 0,5 \text{ Hz}$, $\alpha = 0,94$ (94%).
- ✓ Avec une fréquence d'échantillonnage $f_{\text{ech}} = 250 \text{ Hz}$ et une fréquence de coupure $f_c = 0,5 \text{ Hz}$, $\alpha = 0,987$ (99%).

11.3.2 Programmation du filtre complémentaire

La programmation de ce genre de filtre est relativement simple et l'implémentation se fait donc par le calcul direct :

$$\theta_{\text{fusionné}} = \alpha \cdot \theta_{\text{gyromètre}} + (1 - \alpha) \cdot \theta_{\text{accéléromètre}}$$

Il est souvent conseillé d'utiliser un coefficient α compris entre 96% et 99%.

Sinon comme précisé précédemment on peut calculer le coefficient α :

- ✓ à partir de la période d'échantillonnage Δt et la constante de temps τ du filtre : $\alpha = \frac{\tau}{\tau + \Delta t}$
- ✓ à partir des fréquences d'échantillonnage et de coupure : $\alpha = \frac{f_{\text{ech}}}{f_{\text{ech}} + 2\pi f_c}$

$$\Delta t = \frac{1}{f_{\text{ech}}} \quad \tau = \frac{1}{2\pi f_c}$$

Avec :

Δt : période d'échantillonnage

τ : Constante de temps du filtre

f_c : fréquence de coupure

La fréquence de coupure doit être, théoriquement, inférieure à celle du bruit de l'accéléromètre.

A FAIRE: Implémenter ce filtre en l'insérant dans la structure conditionnelle de calcul périodique.

A FAIRE: Ajouter les commandes pour afficher les angles fusionnés.

A FAIRE: Tester votre code pour évaluer l'amélioration des calculs au regard des problématiques rencontrées.

A FAIRE: Conclure sur l'intérêt et les performances de ce filtre.

11.4 L'algorithme de filtrage de MADGWICK

L'algorithme de Sébastien Madgwick développé pour un travail de recherche « An efficient orientation filter for inertial and inertial/magnetic sensor arrays » est maintenant disponible en open source sur le [site de XIO Technologie](#) qui commercialise aussi des centrales inertielles.

Connu sous le nom d'algorithme AHRS de Madgwick il utilise un algorithme de **descente de gradient** qui permet d'optimiser une fonction réelle différentiable.

Le schéma ci-contre présente le principe de l'algorithme de fusion :

Une bibliothèque est disponible sur le [dépôt Github](#).

Cette bibliothèque utilise l'orientation par les quaternions et pourra calculer des angles d'attitude mais pas nécessairement paramétrés comme dans ce dossier !

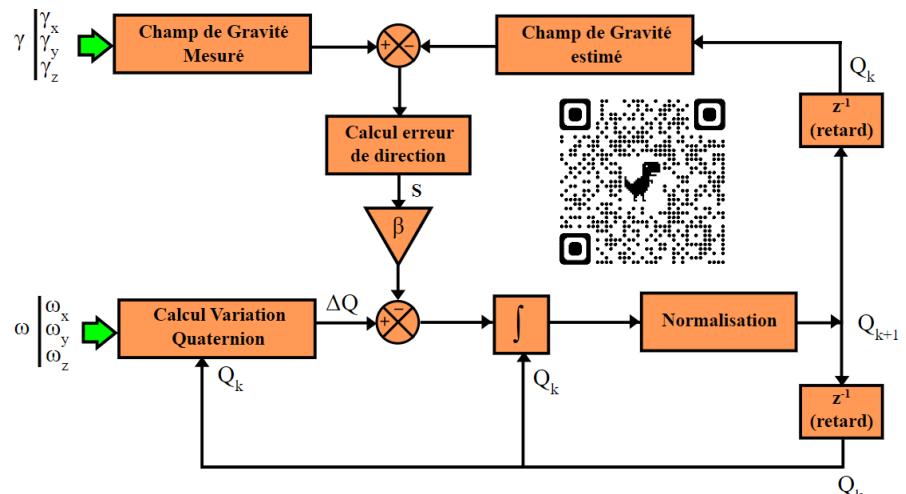


Illustration Vincent Kerhoas - www.enib.fr

L'intégration dans le code arduino se fait grâce aux instructions suivantes :

```
#include <MadgwickAHRS.h>
```

```
Madgwick NOM_DU_FILTRE ;
```

```
NOM_DU_FILTRE.begin( fréquence ) ;
```

Permet d'intégrer la bibliothèque nécessaire qui doit être, au préalable, installée (Menu « Croquis » / « Inclure une bibliothèque » / « Ajouter la bibliothèque .ZIP »)

Activation de la fonction de filtrage de madgwick, au début du programme

Lancement de la procédure à une fréquence définie en Hz à intégrer dans la section < void setup()>

```
NOM_DU_FILTRE.updateIMU(gx, gy, gz, ax, ay, az);
```

Activation du filtre en lui renseignant les 6 valeurs de la centrale (Gyromètre puis accéléromètre)

```
angle = NOM_DU_FILTRE.getPitch();
```

Lecture de la valeur de l'angle de tangage (Pitch)

```
angle = NOM_DU_FILTRE.getRoll();
```

Lecture de la valeur de l'angle de roulis (Roll)

```
angle = NOM_DU_FILTRE.getYaw();
```

Lecture de la valeur de l'angle de lacet (Yaw)

A FAIRE: Implémenter ce filtre à votre programme (vous pouvez conserver le filtre complémentaire afin de les comparer)

A FAIRE: Vérifier les angles calculés. Est-ce que les 3 angles correspondent bien à la définition retenue ?

A FAIRE: Vérifier les performances de ce filtre et conclure sur son intérêt par rapport au filtre complémentaire.

A FAIRE: Conclure sur les limites de ces méthodes et les éventuelles solutions pour de meilleures performances.

11.5 Le filtre de KALMAN

Le filtre de Kalman est un [filtre à réponse impulsionnelle INFINIE](#) (RII), simplifié, qui repose sur l'algorithme suivant :

- ✓ calcul d'un modèle qui hypothétise la prochaine valeur de sortie du capteur,
- ✓ comparaison de la valeur mesurée avec sa supposition,
- ✓ mise à jour du modèle pour optimiser ses prochaines hypothèses.

On parle d'estimateur récursif de la **covariance** du capteur.

Il existe plusieurs variantes de ce filtre dont la plus utilisée est le filtre étendu (EKF) qui, lui aussi, se retrouve sous différentes formes et algorithmes.

Devant la complexité même de la présentation des outils mathématiques pour décrire ce filtre, il n'a pas été détaillé ici et pourra difficilement faire l'objet d'une étude poussée en prébac.

Il existe plusieurs bibliothèques Arduino et exemples de codage de filtres de Kalman que l'on pourra tester sur un microcontrôleur comme celles de [Kristian Lauszus - TKJ Electronics](#) ou [Korneliusz Jarzębski](#).

Pour approfondir le sujet :

[L'estimateur de Kalman Appliqué à la navigation inertie philpetitpa.pagesperso-orange.fr](#)

[Filtre de Kalman - NICOLAS PETIT - Centre Automatique et Systèmes Unité Mathématiques et Systèmes MINES ParisTech](#)

Je laisse le soin au lecteur de réaliser des test sur un arduino équipé d'un microcontrôleur 8 bits et d'ensuite essayer avec une version plus puissante...

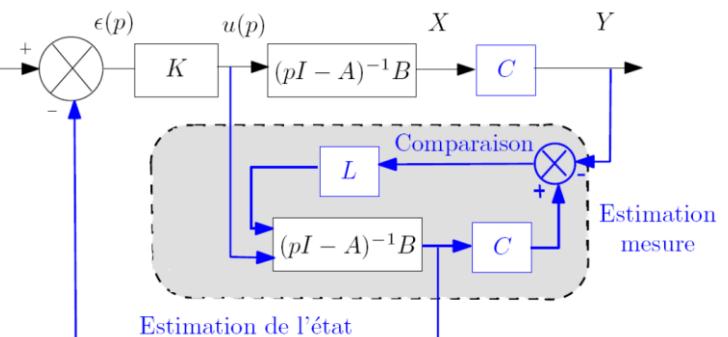


Illustration NICOLAS PETIT – Ecole polytechnique

12 AHRS: ATTITUDE AND HEADING REFERENCE SYSTEMS

Il est évident que ce genre de filtrage ne sera pas optimal et son utilisation avec un Arduino équipé d'un µC 8 bits tournant à 16Mhz sera très limité.

Il serait intéressant de tester ces algorithmes sur un µC plus puissant comme les dernières versions d'arduino équipés d'ARM 32 bits.

Pour aller plus loin dans les connaissances des filtrages, le site ahrs.readthedocs.io propose un ensemble assez exhaustif de ressources et surtout de codes python pour mettre en place et expérimenter **la majorité des différents algorithmes de fusion de données**.

Ce site est une véritable mine d'explications détaillées pour approfondir le domaine.

Il est conseillé de le parcourir pour envisager l'étendue des concepts liés à la mesure d'attitude d'un solide dans l'espace.

Pour aller plus loin sur le sujet voici quelques liens vers des ressources intéressantes :

[Le problème de Wahba et quelques algorithmes de résolution A. Garcia, F. Laurent, F-X. Morel March 30, 2010](#)

[Hassen Fourati, Noureddine Manamanni, Yves Handrich. Fusion de données, estimation de la posture et navigation à l'estime : Application au Bio-logging. Editions Universitaires Européennes, pp.208, 2012, Mathématiques, 978-3-8417-8971-6. fhal-00690144f](#)

[Fourati, Hassen & Manamanni, Noureddine & Afilal, Lissan & Handrich, Yves. \(2010\). Reconstitution de l'orientation et des accélérations corporelles à l'aide de capteurs inertiel et magnétiques.](#)

[Fermi Guerrero Castellanos. Estimation de l'attitude et commande bornée en attitude d'un corps rigide : application à un mini hélicoptère à quatre rotors. Automatique / Robotique. Université Joseph-Fourier - Grenoble I. 2008. Français. ffnnt : ff. fftel-00370222f](#)

[Fusion de données multicapteurs pour la capture de mouvement Bernardino Benito Salmeron-Quiroz](#)

[Estimation d'attitude et diagnostic d'une centrale d'attitude par des outils ensemblistes Hoang Van Nguyen](#)

13 LE BLOCAGE DE CARDAN

Les calculs précédents, mais il en serait de même avec les matrices de rotation, présentent un **problème de singularités** connu sous le nom de **blocage de cardan**.

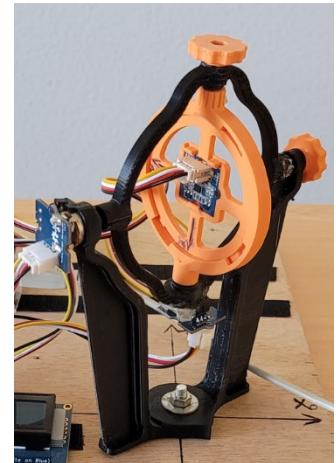
C'est un problème connu depuis très longtemps (voir anecdote sur la mission Apollo) qui est présent dans tous les gyroscopes.

Il ne s'agit pas réellement d'un blocage mais d'un problème de calcul lorsque les angles sont proches de $\pm 90^\circ$.

Comme dans l'exemple ci-contre, l'axe transversal \vec{Y}_{IMU} (donc de tangage) de la centrale se retrouve, colinéaire avec l'axe de Lacet \vec{Y}_0 .

Un des angles est de ce fait indéfini il ne sera donc « pas calculable » ou, tout au moins, sa valeur risque d'être faussée.

Le calcul des angles par l'accéléromètre sera perturbé avec le risque de confusion de certaines valeurs.



13.1 Utilisation des quaternions

Pour éviter ce problème il est possible d'utiliser une représentation de l'attitude avec les **quaternions** qui sont très largement utilisés pour représenter les rotations spatiales dans :

- ✓ le contrôle en robotique,
- ✓ le contrôle en aéronautique (satellites et fusées),
- ✓ mais aussi, *plus surprenant*, dans les **jeux vidéo**.

Les démonstrations mathématiques sur les quaternions ne sont pas facilement accessibles, encore moins pour un élève de pré-bac.

Mais cet outil mathématique est très présent et donc très utilisé dans les publications sur le thème de l'attitude d'un solide dans l'espace.

Les explications ci-dessous évitent volontairement de détailler les démonstrations utilisant leurs propriétés particulières et **VULGARISE LEUR NOTATION** afin de la rendre la plus lisible possible par un néophyte.

Ils ne sont donc présentés ici, que pour la découverte de l'outil utilisé dans certains algorithmes de filtrages et, dans un premier temps, il est tout à fait possible de les ignorer.

Pour une meilleure compréhension de l'outil quaternion, il convient de se plonger dans ces concepts qui demandent un temps et un effort certain pour appréhender l'abstraction utilisée.

Vous trouverez ci-dessous, une liste non exhaustive de liens vers des ressources issues du monde universitaire plus ou moins facile d'approche.

[Le blog de LULU – quaternion et gyroscope](#)

[A la découverte des hypercomplexes - Fractales67 - complexe.jimdo.com](#)

[Comparaisons de performances avec d'autres méthodes de rotation - wikipedia](#)

[Théorie des quaternions \(Hamilton, 1843\) - François NICOLASr](#)

[Florian Monteghetti. Quaternions, orientation et mouvement. \[Rapport de recherche\] ISAESUPAERO. 2012. fffhal-01618257f](#)

[Système de Repérage 3D pour Caméra - Cedric Lienhard – HEIG-VD - Département des Technologies Industrielles Filière Microtechnique](#)

[ROTATION: A review of useful theorems involving proper orthogonal matrices referenced to threedimensional physical space](#)

13.1.1 Le quaternion unitaire

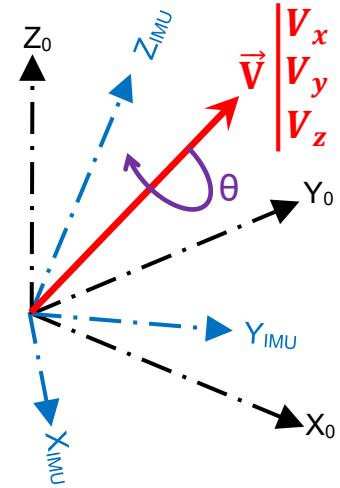
La « découverte » des quaternions par Carl Friedrich Gauss et William Rowan Hamilton remonte au 19^{ème} siècle. Cette représentation, appelée aussi **paramètre symétrique d'Euler** fut rapidement utilisé pour répondre au problème de Wahba.

En vulgarisant, il s'agit d'un « nombre hypercomplexe à 4 dimensions » qui peut être décrit par un « vecteur » **Q** à quatre composantes :

$$Q = [q_\omega, q_x, q_y, q_z]$$

Plutôt que d'utiliser 3 rotations successives, l'orientation du référentiel **R_{IMU}** par rapport à **R₀** se représente par un **quaternion unité** $Q_{R_{IMU} \rightarrow R_0}$, normalisé.

Ce quaternion, aussi appelé axe d'Euler, est composé d'un vecteur \vec{V} autour duquel s'effectue une rotation d'un angle θ .



Il peut être défini comme suit :

$$Q_{R_{IMU} \rightarrow R_0} = \left[\cos \frac{\theta}{2}, \frac{V_x \cdot \sin \frac{\theta}{2}}{\sqrt{V_x^2 + V_y^2 + V_z^2}}, \frac{V_y \cdot \sin \frac{\theta}{2}}{\sqrt{V_x^2 + V_y^2 + V_z^2}}, \frac{V_z \cdot \sin \frac{\theta}{2}}{\sqrt{V_x^2 + V_y^2 + V_z^2}} \right]$$

Pour une position donnée, il n'existe qu'un seul quaternion unité représentant l'attitude.

AVANTAGES

- Les quaternions « ne sont pas » soumis au blocage de cardan
- L'écriture mathématique des équations est simplifiée (si l'outil quaternion est maîtrisé)

INCONVENIENTS

- Il est difficile de visualiser l'orientation réelle du solide avec cet outil,
- la maîtrise de l'outil mathématique est assez complexe,
- le calcul des angles de LACET, ROULIS et TANGAGE à partir des quaternions n'est « naturel ».

13.1.2 Le quaternion « vecteur »

La centrale donne les composantes des vecteurs accélérations \vec{a}_{IMU} et vitesses angulaires $\vec{\omega}_{IMU}$ dans la base **R_{IMU}** de la centrale. Il est possible de représenter 2 quaternions correspondant chacun à son vecteur :

$$Q_{\vec{\omega}_{R_{IMU}}} = [0, \omega_x, \omega_y, \omega_z] \quad \text{et} \quad Q_{\vec{a}_{R_{IMU}}} = [0, a_x, a_y, a_z]$$

13.1.3 Changement de base d'un quaternion vecteur

Il est possible de calculer un quaternion vecteur $Q_{\vec{a}_{R_0}}$ dans la base du référentiel fixe **R₀** à partir du quaternion vecteur $Q_{\vec{a}_{R_{IMU}}}$ mesuré par l'accéléromètre dans le référentiel du capteur ainsi que du quaternion $Q_{R_{IMU} \rightarrow R_0}$ de rotation entre les 2 référentiels.

$$Q_{\vec{a}_{R_0}} = Q_{R_{IMU} \rightarrow R_0} \otimes Q_{\vec{a}_{R_{IMU}}} \otimes Q_{R_{IMU} \rightarrow R_0}^{-1}$$

Le produit \otimes est une « multiplication associative » qui correspond plus à un produit vectoriel. Cette opération est équivalente à l'utilisation d'une matrice de rotation entre les 2 vecteurs.

$Q_{R_{IMU} \rightarrow R_0}^{-1}$, souvent noté $\overline{Q_{R_{IMU} \rightarrow R_0}}$ est le quaternion inverse soit $Q_{R_0 \rightarrow R_{IMU}}$

13.1.4 Du quaternion aux angles d'Euler

Il est possible de définir des « formules » de passage de l'un à l'autre

Quaternion unité à partir des angles d'Euler :

$$\begin{aligned} q_\omega &= \pm \left[\cos\left(\frac{\varphi}{2}\right) \cos\left(\frac{\theta}{2}\right) \cos\left(\frac{\psi}{2}\right) + \sin\left(\frac{\varphi}{2}\right) \sin\left(\frac{\theta}{2}\right) \sin\left(\frac{\psi}{2}\right) \right] \\ q_x &= \pm \left[\sin\left(\frac{\varphi}{2}\right) \cos\left(\frac{\theta}{2}\right) \cos\left(\frac{\psi}{2}\right) - \cos\left(\frac{\varphi}{2}\right) \sin\left(\frac{\theta}{2}\right) \sin\left(\frac{\psi}{2}\right) \right] \\ q_y &= \pm \left[\cos\left(\frac{\varphi}{2}\right) \sin\left(\frac{\theta}{2}\right) \cos\left(\frac{\psi}{2}\right) + \sin\left(\frac{\varphi}{2}\right) \cos\left(\frac{\theta}{2}\right) \sin\left(\frac{\psi}{2}\right) \right] \\ q_z &= \pm \left[\cos\left(\frac{\varphi}{2}\right) \cos\left(\frac{\theta}{2}\right) \sin\left(\frac{\psi}{2}\right) - \sin\left(\frac{\varphi}{2}\right) \sin\left(\frac{\theta}{2}\right) \cos\left(\frac{\psi}{2}\right) \right] \end{aligned}$$

Angles d'Euler à partir du quaternion :

$$\varphi = \arctan\left(\frac{2(q_\omega q_x + q_y q_z)}{q_\omega^2 + q_z^2 - q_x^2 - q_y^2}\right)$$

$$\theta = \arcsin\left(2(q_\omega q_y - q_x q_z)\right)$$

$$\psi = \arctan\left(\frac{2(q_\omega q_z + q_x q_y)}{q_\omega^2 + q_x^2 - q_y^2 - q_z^2}\right)$$

Ces calculs seront différents suivant le choix de l'ordre des rotations présenté au [chapitre paramétrage](#).

Afin de visualiser ces concepts vous pouvez accéder à 2 simulations « 3D » correspondant à ces calculs :

[Animation 3D du passage des quaternions aux angles d'Euler](#)

[Simulation des angles d'Euler - quaternions](#)

13.1.5 Quaternion d'attitude à partir du GYROMETRE

Pour obtenir le quaternion d'attitude Q_{gyro} il faut réaliser [une intégration numérique](#) du quaternion de vitesse angulaire \dot{Q}_{gyro} à intervalles Δt réguliers en utilisant le quaternion $Q_{\bar{\omega}_{R_0}}$:

$$\dot{Q}_{gyro} = \frac{1}{2} \cdot Q_{gyro_ précédent} \otimes Q_{\bar{\omega}_{R_0}} \quad \text{et} \quad Q_{gyro} = Q_{gyro \text{ précédent}} + \dot{Q}_{\omega_{R_0}} \cdot \Delta t$$

13.1.6 Quaternion d'attitude à partir de l'ACCELEROMETRE

Plus complexe à mettre en œuvre, il s'agit de calculer le quaternion d'attitude à partir des composantes du vecteur accélération mais ce calcul nécessite un traitement numérique :

[Valenti RG, Dryanovski I, Xiao J. Keeping a Good Attitude: A Quaternion-Based Orientation Filter for IMUs and MARGs Sensors \(Basel\). 2015 Aug;6\(15\):19302-30. doi: 10.3390/s150819302. PMID: 26258778; PMCID: PMC4570372.](#)

$$Q_{acc} = \begin{cases} \left[-\frac{a_y}{\sqrt{2(1-a_z)}}, \sqrt{\frac{1-a_z}{2}}, 0, \frac{a_x}{\sqrt{2(1-a_z)}} \right] & \text{si } a_z \geq 0 \\ \left[\frac{a_y}{\sqrt{2(1-a_z)}}, \sqrt{\frac{1-a_z}{2}}, 0, -\frac{a_x}{\sqrt{2(1-a_z)}} \right] & \text{si } a_z \leq 0 \end{cases}$$

Ce calcul se retrouve assez rarement dans la bibliographie qui reprend généralement d'autres méthodes comme par exemple :

- ✓ calculer le quaternion Q d'attitude à partir du gyromètre Q_{gyro} seul,
- ✓ déterminer le quaternion $Q_{\bar{a}_{R_0}} - Q_{g_{norm}}$ (quaternion gravité normalisé),
- ✓ corriger le calcul du quaternion d'attitude Q avec cette comparaison.

La correction se fait aussi souvent à partir de la mesure avec un magnétomètre intégré directement dans un algorithme de fusion des capteurs comme expliqué dans le chapitre sur la fusion de capteurs.

en convention NED :

$$\vec{g} \begin{bmatrix} 0 \\ 0 \\ 9.81 \end{bmatrix}$$

En normalisant :

$$\vec{g}_{normalisé} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

14 CONCLUSION

Cette formation n'a pas l'ambition d'apprendre la maîtrise exhaustive de toutes les théories de calculs d'attitude à partir d'une centrale inertielle.

Les expérimentations proposées permettent de découvrir :

- ✓ les limites du calcul des valeurs mesurées à partir de la centrale,
- ✓ les limites du calcul de l'attitude à partir de l'accéléromètre et du gyromètre,
- ✓ les limites de la fusion des capteurs à partir d'algorithmes de filtrages « légers ».

Pour approfondir le sujet il resterait de nombreuses expérimentations à mener concernant :

- ✓ les réglages des filtres internes de la puce pour optimiser les mesures,
- ✓ l'utilisation d'outils mathématiques plus poussés pour éviter le problème du blocage de cardan avec les calculs à partir de l'accéléromètre,
- ✓ l'utilisation des timers du microcontrôleur pour réduire les effets de dérive sur l'intégration numérique dans les calculs à partir du gyromètre,
- ✓ l'étude des algorithmes de fusion plus performants,
- ✓ ...

L'utilisation d'un Arduino UNO équipé d'un µC ATMEGA 328p aux performances limitées reste, bien entendu, un frein à l'utilisation d'algorithmes plus efficaces.

L'arrivée récentes de dernières versions équipées d'un µC ARM 32bits et aux fréquences beaucoup plus élevées pourrait permettre de meilleures performances mais je n'ai pas encore pu tester la chose.

Pour approfondir le sujet, l'étude du site ahrs.readthedocs.io est préconisé, même s'il demande un effort certain pour comprendre les abstractions mathématiques des différents algorithmes proposés.

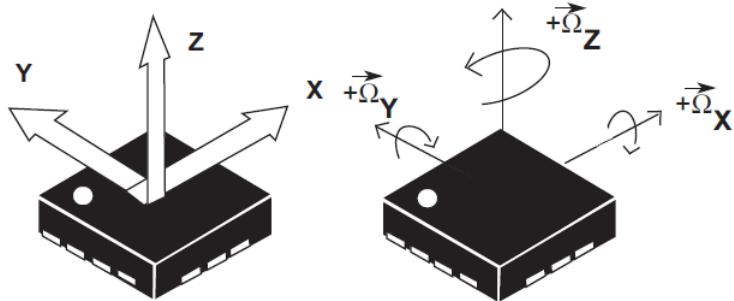
Il propose de nombreuses bibliothèques python facilitant l'implémentation des principaux algorithmes de filtrage et donc leurs tests sur des µC plus performants.

15 LES REGISTRES DE REGLAGES DE LA CENTRALE

Les réglages présentés dans cette section correspondent à ceux du module Grove **IMU 6 DOF** qui intègre une puce « **LSM6DS3** » commercialisée par la société STMicroelectronics mais on retrouve les mêmes informations dans tous les IMU.

15.1 Axes de la centrale

Les axes sont définis dans la datasheet mais ils peuvent être parfois modifiés par un réglage d'un registre.



15.2 Caractéristiques mécaniques

Table 3. Mechanical characteristics

Symbol	Parameter	Test conditions	Min.	Typ. ⁽¹⁾	Max.	Unit
LA_FS	Linear acceleration measurement range			±2		g
				±4		
				±8		
				±16		
G_FS	Angular rate measurement range		±125			dps
			±245			
			±500			
			±1000			
			±2000			
LA_So	Linear acceleration sensitivity	FS = ±2	0.061			mg/LSB
		FS = ±4	0.122			
		FS = ±8	0.244			
		FS = ±16	0.488			
G_So	Angular rate sensitivity	FS = ±125	4.375			mdps/LSB
		FS = ±245	8.75			
		FS = ±500	17.50			
		FS = ±1000	35			
		FS = ±2000	70			
LA_SoDr	Linear acceleration sensitivity change vs. temperature ⁽²⁾	from -40° to +85° delta from T=25°		±1		%
G_SoDr	Angular rate sensitivity change vs. temperature ⁽²⁾	from -40° to +85° delta from T=25°		±1.5		%
LA_TyOff	Linear acceleration typical zero-g level offset accuracy ⁽³⁾			±40		mg
G_TyOff	Angular rate typical zero-rate level ⁽³⁾			±10		dps
LA_OffDr	Linear acceleration zero-g level change vs. temperature ⁽²⁾			±0.5		mg/°C
G_OffDr	Angular rate typical zero-rate level change vs. temperature ⁽²⁾			±0.05		dps/°C
Rn	Rate noise density			7		mdps/√Hz
An	Acceleration noise density	FS = ±2 g ODR = 104 Hz		90		µg/√Hz

Les principales caractéristiques sont :

- ✓ La valeur pleine échelle :
 - Accéléromètre : ± 2 / 4 / 8 / 16 g
 - Gyromètre : ± 125 / 245 / 500 / 1000 / 2000 °/s
- ✓ Les bandes passantes (en Hz)
- ✓ Les sensibilités en mg/LSB ou mdps/LSB
- ✓ La gamme de température d'utilisation (en °C) et la variation de sensibilité en LSB/°C.

Table 3. Mechanical characteristics (continued)

Symbol	Parameter	Test conditions	Min.	Typ. ⁽¹⁾	Max.	Unit
LA_ODR	Linear acceleration output data rate			13 26 52 104 208 416 833 1666 3332 6664		Hz
G_ODR	Angular rate output data rate			13 26 52 104 208 416 833 1666		
Top	Operating temperature range		-40		+85	°C

1. Typical specifications are not guaranteed.

2. Measurements are performed in a uniform temperature setup.

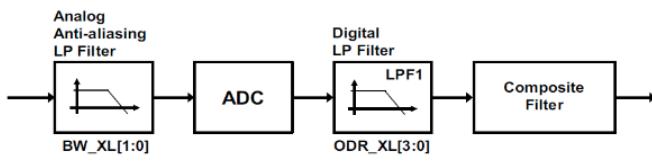
3. Values after soldering.

15.3 Les fonctionnalités

Les centrales intègrent de nombreuses fonctionnalités parfaitement documentées dans la datasheet comme :

- ✓ les interfaces de communication : **i2C**, SPI ;
- ✓ l'activation / désactivation possible de l'accéléromètre et / ou du gyromètre ;
- ✓ le contrôle du mode de performance : « Power-down », « low-power », « normal » ou « hight performance » ;
- ✓ Un mode de fonctionnement avec une mémoire tampon de type file (FIFO) qui est utile pour limiter les consommations dans certaines applications ;
- ✓ des **filtres** réglables dans la chaîne numérique de traitement de chaque composant ;

Chaine de traitement numérique de l'accéléromètre



Chaine de traitement numérique du gyromètre

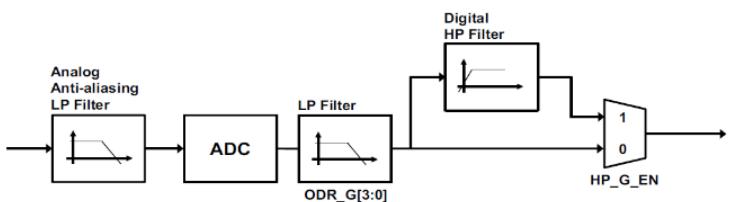
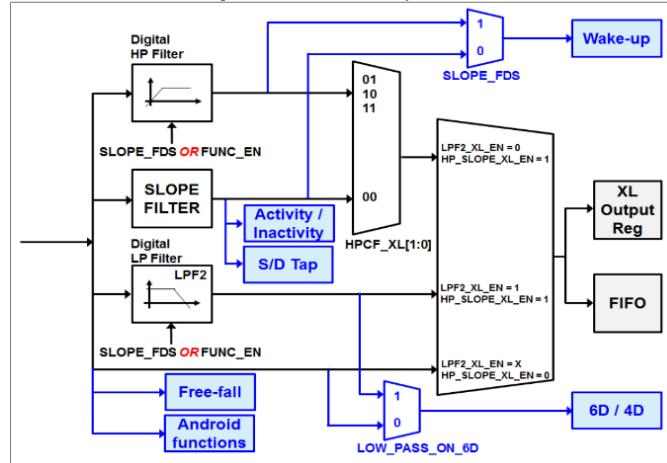


Figure 5. Accelerometer composite filter



A noter que les centrales inertielles possèdent aussi des filtres passe haut et passe bas qui peuvent être contrôlés précisément en fonction des applications mais ne sont pas présentés dans cette formation.

- ✓ des **fonctions** particulières comme la détection de chute ou de mouvement qui sont activables pour des applications particulières ;

15.4 Les Principaux REGISTRES

Toutes ces fonctionnalités sont parfaitement réglables grâce aux registres qui intègrent aussi les réglages des valeurs pleine échelle.

- ✓ CTRL1_XL (10h) : réglage de l'accéléromètre.
- ✓ CTRL2_G (11h) : réglage du gyromètre.
- ✓ CTRL3_C (12h) : réglage du BLE

L'ensemble des registres utiles aux réglages est détaillé dans l'[ANNEXE1](#) en fin de dossier.

Name	Type	Register address		Default	Comment
		Hex	Binary		
FUNC_CFG_ACCESS	r/w	01	00000001	00000000	Embedded functions configuration register
ORIENT_CFG_G	r/w	0B	00001011	00000000	
CTRL1_XL	r/w	10	00010000	00000000	
CTRL2_G	r/w	11	00010001	00000000	
CTRL3_C	r/w	12	00010010	00000100	
CTRL4_C	r/w	13	00010011	00000000	
CTRL5_C	r/w	14	00010100	00000000	
CTRL6_C	r/w	15	00010101	00000000	
CTRL7_G	r/w	16	00010110	00000000	
CTRL8_XL	r/w	17	00010111	00000000	
CTRL9_XL	r/w	18	00011000	00111000	
CTRL10_C	r/w	19	00011001	00111000	

16 LECTURE DES VALEURS AVEC LES REGISTRES

Cette expérimentation propose de découvrir, pas à pas, l'acquisition des valeurs à partir des registres.

Si les difficultés de programmation C vous rebutent, passez directement au chapitre suivant...

16.1 Communication I2C avec la centrale

Les centrales numériques proposent généralement une communication SPI ou I2C. Le choix retenu ici est celui de l'I2C qui est facilement implantable avec un Arduino grâce à la bibliothèque **wire.h**, livrée par défaut avec l'IDE.

Pour info, l'Arduino UNO utilise 2 broches pour la communication I2C : A4 (SDA) et A5 (SCL) mais en utilisant le Shield GROVE il suffit d'utiliser l'un des 4 connecteurs dédiés.

Les premiers codes fournis ci-dessous sont utilisés pour la lecture ou l'écriture des registres de la centrale et nécessitent :

- ✓ l'importation de la bibliothèque en début de programme (déjà vu dans les programmes précédents) ;
- ✓ la déclaration du démarrage de la communication, **dans le setup** ;
- ✓ une **fonction ECRITURE_REGISTRES()** pour modifier un registre de la centrale ;
- ✓ une **fonction LECTURE_REGISTRES()** pour lire x registres de la centrale.

*Les fonctions ECRITURE et LECTURE utilisent les paramètres : Add_module, Add_Registre, Nregistres et * pTableau décrits dans le programme de lecture des valeurs brutes.*

```
#include <Wire.h> //Intégration de la bibliothèque "wire" pour la gestion de l'I2C

void setup()
{
    Wire.begin();           //lancement de la fonction communication
}

void LECTURE_REGISTRES( uint8_t Add_module, uint8_t Add_Registre, uint8_t Nregistres,
uint8_t * pTableau )
{
    //Initialisation de la connexion avec le module (Adresse : du module)
    Wire.beginTransmission(Add_module);
    //Définition de l'adresse du PREMIER registre
    Wire.write(Add_Registre);
    //Arrêt de l'envoie d'informations sur l'I2C
    Wire.endTransmission();
    //Requette de lecture des n octets
    Wire.requestFrom(Add_module, Nregistres);
    //initialisation d'une variable de comptage
    uint8_t index=0;
    //Boucle Tant que des octets sont disponibles, enregistrer la valeur de chaque octet dans le tableau
    et incrémenter l'index
    while ( Wire.available() ) pTableau[index++] = Wire.read();
}

void ECRITURE_REGISTRE( uint8_t Add_module, uint8_t Add_Registre, uint8_t Valeur )
{
    //Initialisation de la connexion avec le module (Adresse : du module)
    Wire.beginTransmission(Add_module);
    //Définition de l'adresse du registre
    Wire.write(Add_Registre);
    //Ecriture de la valeur
    Wire.write(Valeur);
    //Arrêt de l'envoie d'informations sur l'I2C
    Wire.endTransmission();
}
```

16.2 Réglages de la centrale par les registres

Avant de pouvoir l'utiliser, il est nécessaire de procéder aux réglages de la centrale, en particulier les valeurs pleine échelle de l'accéléromètre ($\pm 2 / 4 / 8 / 16 \text{ g}$) et du gyromètre ($\pm 125 / 245 / 500 / 1000 / 2000 \text{ }^{\circ}/\text{s}$).

Le réglage proposé, correspondant à $\pm 16\text{g}$ et $\pm 2000 \text{ }^{\circ}/\text{s}$, est intégré dans une fonction **reglage_LM6DS3()** qu'il convient d'appeler dans la routine **setup()**.

```
void reglage_LM6DS3()
{
    /* -- REGISTRE : FUNC_CFG_ACCESS (01h) : Désactive (/active) les fonctions embarquées */
    ECRITURE_REGISTRE( 0x6A , 0x01 , 0b00000000 );

    /* -- REGISTRE : ORIENT_CFG_G (0Bh) Angular rate sensor sign and orientation register */
    ECRITURE_REGISTRE( 0x6A , 0x0B , 0b00000000 );

    /* -- REGISTRE : CTRL1_XL (10h) Linear acceleration sensor control register 1 */
    ECRITURE_REGISTRE( 0x6A , 0x10 , 0b01100100 ); /* ±16g et autres réglages par défaut */

    /* -- REGISTRE : CTRL2_G (11h) Angular rate sensor control register 2 */
    ECRITURE_REGISTRE( 0x6A , 0x11 , 0b01101100 ); /* ±2000°/s et autres réglages par
défaut */

    /* -- REGISTRE : CTRL3_C (12h) Control register 3 */
    ECRITURE_REGISTRE( 0x6A , 0x12 , 0b00000010 ); /* activation du BLE MSB @ Lower adress */

    /* -- REGISTRE : CTRL4_C (13h) Control register 4 */
    ECRITURE_REGISTRE( 0x6A , 0x13 , 0b10000000 );

    /* -- REGISTRE : CTRL5_C (14h) Control register 5 */
    ECRITURE_REGISTRE( 0x6A , 0x14 , 0b00000000 );

    /* -- REGISTRE : CTRL6_C (15h) Control register 6 */
    ECRITURE_REGISTRE( 0x6A , 0x15 , 0b00000000 );

    /* -- REGISTRE : CTRL7_G (16h) Angular rate sensor control register 7 */
    ECRITURE_REGISTRE( 0x6A , 0x16 , 0b00000000 );

    /* -- REGISTRE : CTRL8_XL (17h) Linear acceleration sensor control register 8 */
    ECRITURE_REGISTRE( 0x6A , 0x17 , 0b00000000 );

    /* -- REGISTRE : CTRL9_XL (18h) Linear acceleration sensor control register 9 (r/w) */
    ECRITURE_REGISTRE( 0x6A , 0x18 , 0b00111000 );

    /* -- REGISTRE : CTRL10_C (19h) Control register 10 (r/w) */
    ECRITURE_REGISTRE( 0x6A , 0x19 , 0b00111000 );
}
```

Ici j'ai fait le choix de n'appliquer que des réglages simples sans utiliser des filtres intégrés qu'il sera possible d'activer (voir [le chapitre sur les réglages](#) et /ou [l'annexe 1 : les principaux registres de la centrale](#)).

Pour les centrales inertielles de STMicroelectronics, une fonction **reglage_MPU9250()** est proposée dans les fichiers d'accompagnement.

16.3 Lecture des valeurs brutes

Comme expliqué au [chapitre sur la lecture des valeurs brutes de la centrale](#), chaque valeur mesurée par la centrale est stockée sur 16 bits, donc dans 2 registres différents **High** (poids fort) et **Low** (poids faible).

Il faut donc lire les 2 registres séparément et réaliser une **addition binaire** pour retrouver la valeur brute qui consiste à :

- ✓ lire l'octet de poids fort,
- ✓ réaliser un décalage de 8 bits vers la gauche ($<< 8$),
- ✓ additionner (+) avec l'octet de poids faible.

Le code proposé permet, pour chaque capteur, :

- ✓ d'appeler la fonction `LECTURE_REGISTRES()` pour stocker les 6 octets successifs dans un tableau,
- ✓ de réaliser l'addition binaire pour chaque axe et stocker la valeur dans une variable indépendante.

Name	Register adress (Hex)	Comment
OUT_TEMP_L	20	Température output data register
OUT_TEMP_H	21	
OUTX_L_G	22	
OUTX_H_G	23	
OUTY_L_G	24	Gyroscope output data register
OUTY_H_G	25	
OUTZ_L_G	26	
OUTZ_H_G	27	
OUTX_L_XL	28	
OUTX_H_XL	29	
OUTY_L_XL	2A	
OUTY_H_XL	2B	
OUTZ_L_XL	2C	
OUTZ_H_XL	2D	Accelerometer output data register

```
//Variables pour la lecture des valeurs brutes d'accéléromètre
    int16_t      ax_brut, ay_brut, az_brut;
//Variables pour la lecture des valeurs brutes des vitesses angulaires (gyromètre)
    int16_t      gx_brut, gy_brut, gz_brut;

void lecture_valeurs_brutes()
{
    //Variable locale pour déclarer un pointeur vers un tableau dynamique de 6 valeurs
    uint8_t pTampon[6];

    LECTURE_REGISTRES( 0x6A , ##### , 6 , pTampon );
    ax_brut=(pTampon[0] << 8 | pTampon[1] );
    ay_brut=(pTampon[2] << 8 | pTampon[3] );
    az_brut=(pTampon[4] << 8 | pTampon[5] );

    LECTURE_REGISTRES( 0x6A , ##### , 6 , pTampon );
    gx_brut=( pTampon[0] << 8 | pTampon[1] );
    gy_brut=( pTampon[2] << 8 | pTampon[3] );
    gz_brut=( pTampon[4] << 8 | pTampon[5] );
}
```

Attention l'utilisation du réglage du BLE dans le registre **CTRL3_C** (12h) inverse les octets de poids faible et de poids forts. Ce qui implique ici que le premier octet lu est bien celui de poids fort contrairement à ce qui est indiqué dans la datasheet...

Les fonctions **ECRITURE** et **LECTURE** utilisent les paramètres :

- ✓ **Add_module** : adresse I2C du module ici 0x6A pour la centrale LSM6DS3 (ou 0x68 pour les centrales InvenSense TDK MPU9250 et ICM20600),
- ✓ **Add_Registre** : adresse du registre du composant à modifier ou du 1^{er} registre à lire,
- ✓ **Nregistres** : Nombre de registres successifs à lire (pour la lecture uniquement)
- ✓ *** pTableau** : « Pointeur » vers l'adresse de stockage du tableau fourni en paramètre.

A FAIRE: Retrouver les adresses des registres de lecture des valeurs de l'accéléromètre et du gyromètre et compléter l'appel des fonctions dans le code.

17 ANNEXE 1 :REGISTRES DE REGLAGES DU LSM6DS3

Registre FUNC_CFG_ACCESS (01h) : Désactive (/active) les fonctions embarquées

Table 17. FUNC_CFG_ACCESS register

FUNC_CFG_EN	0 ⁽¹⁾						
-------------	------------------	------------------	------------------	------------------	------------------	------------------	------------------

1. This bit must be set to '0' for the correct operation of the device.

Table 18. FUNC_CFG_ACCESS register description

FUNC_CFG_EN	Enable access to the embedded functions configuration registers ⁽¹⁾ from address 02h to 32h. Default value: 0. (0: disable access to embedded functions configuration registers; 1: enable access to embedded functions configuration registers)
-------------	---

1. The embedded functions configuration registers details are available in 10: Embedded functions register mapping and 11: Embedded functions registers description.

REGLAGE = 0b 0000 0000 désactiver les fonctions

REGISTRE ORIENT_CFG_G (0Bh) Angular rate sensor sign and orientation register

Table 37. ORIENT_CFG_G register

0 ⁽¹⁾	0 ⁽¹⁾	SignX_G	SignY_G	SignZ_G	Orient_2	Orient_1	Orient_0
------------------	------------------	---------	---------	---------	----------	----------	----------

1. This bit must be set to '0' for the correct operation of the device.

Table 38. ORIENT_CFG_G register description

SignX_G	Pitch axis (X) angular rate sign. Default value: 0 (0: positive sign; 1: negative sign)
SignY_G	Roll axis (Y) angular rate sign. Default value: 0 (0: positive sign; 1: negative sign)
SignZ_G	Yaw axis (Z) angular rate sign. Default value: 0 (0: positive sign; 1: negative sign)
Orient [2:0]	Directional user-orientation selection. Default value: 000 For the configuration setting, refer to Table 39.

Table 39. Settings for orientation of axes

Orient [2:0]	000	001	010	011	100	101
Pitch	X	X	Y	Y	Z	Z
Roll	Y	Z	X	Z	X	Y
Yaw	Z	Y	Z	X	Y	X

REGLAGE = 0b 0000 0010

- Sens positif pour chaque axe ;
- ROULIS = axe X, TANGAGE = axe Y et LACET = axe Z

REGISTRE CTRL1_XL (10h) Linear acceleration sensor control register 1

Table 45. CTRL1_XL register

ODR_XL3	ODR_XL2	ODR_XL1	ODR_XL0	FS_XL1	FS_XL0	BW_XL1	BW_XL0
---------	---------	---------	---------	--------	--------	--------	--------

Table 46. CTRL1_XL register description

ODR_XL [3:0]	Output data rate and power mode selection. Default value: 0000 (see Table 47).
FS_XL [1:0]	Accelerometer full-scale selection. Default value: 00. (00: ±2 g; 01: ±16 g; 10: ±4 g; 11: ±8 g)
BW_XL [1:0]	Anti-aliasing filter bandwidth selection. Default value: 00 (00: 400 Hz; 01: 200 Hz; 10: 100 Hz; 11: 50 Hz)

Table 47. Accelerometer ODR register setting

ODR_XL3	ODR_XL2	ODR_XL1	ODR_XL0	ODR selection [Hz] when XL_HM_MODE = 1	ODR selection [Hz] when XL_HM_MODE = 0
0	0	0	0	Power-down	Power-down
0	0	0	1	13 Hz (low power)	13 Hz (high performance)
0	0	1	0	26 Hz (low power)	26 Hz (high performance)
0	0	1	1	52 Hz (low power)	52 Hz (high performance)
0	1	0	0	104 Hz (normal mode)	104 Hz (high performance)
0	1	0	1	208 Hz (normal mode)	208 Hz (high performance)
0	1	1	0	416 Hz (high performance)	416 Hz (high performance)
0	1	1	1	833 Hz (high performance)	833 Hz (high performance)
1	0	0	0	1.66 kHz (high performance)	1.66 kHz (high performance)
1	0	0	1	3.33 kHz (high performance)	3.33 kHz (high performance)
1	0	1	0	6.66 kHz (high performance)	6.66 kHz (high performance)

Table 48. BW and ODR (high-performance mode)

ODR ⁽¹⁾	Analog filter BW (XL_HM_MODE = 0)	
	XL_BW_SCAL_ODR = 0	XL_BW_SCAL_ODR = 1
6.66 - 3.33 kHz	Filter not used	
1.66 kHz	400 Hz	
833 Hz	400 Hz	
416 Hz	200 Hz	
208 Hz	100 Hz	
104 - 13 Hz	50 Hz	

1. Filter not used when accelerometer is in normal and low-power modes.

REGLAGE = 0b 0110 ####0

- ODR = 416 Hz Mode High perf donc XL_HM_MODE = 0 voir reg 15h
- Valeur pleine échelle = paramètre important à régler
- antialiasing = 400 Hz

REGISTRE CTRL2_G (11h) Angular rate sensor control register 2

Table 49. CTRL2_G register

ODR_G3	ODR_G2	ODR_G1	ODR_G0	FS_G1	FS_G0	FS_125	0 ⁽¹⁾
--------	--------	--------	--------	-------	-------	--------	------------------

1. This bit must be set to '0' for the correct operation of the device.

Table 50. CTRL2_G register description

ODR_G [3:0]	Gyroscope output data rate selection. Default value: 0000 (Refer to Table 49)
FS_G [1:0]	Gyroscope full-scale selection. Default value: 00 (00: 245 dps; 01: 500 dps; 10: 1000 dps; 11: 2000 dps)
FS_125	Gyroscope full-scale at 125 dps. Default value: 0 (0: disabled; 1: enabled)

REGLAGE = 0b 0110 ####0

- ODR = 416 Hz Mode High perf donc G_HM_MODE= 0 voir reg 16h
- Valeur pleine échelle = paramètre important à régler

REGISTRE CTRL3_C (12h) Control register 3

Table 52. CTRL3_C register

BOOT	BDU	H_LACTIVE	PP_OD	SIM	IF_INC	BLE	SW_RESET
------	-----	-----------	-------	-----	--------	-----	----------

Table 53. CTRL3_C register description

BOOT	Reboot memory content. Default value: 0 (0: normal mode; 1: reboot memory content ⁽¹⁾)
BDU	Block Data Update. Default value: 0 (0: continuous update; 1: output registers not updated until MSB and LSB have been read)
H_LACTIVE	Interrupt activation level. Default value: 0 (0: interrupt output pads active high; 1: interrupt output pads active low)
PP_OD	Push-pull/open-drain selection on INT1 and INT2 pads. Default value: 0 (0: push-pull mode; 1: open-drain mode)
SIM	SPI Serial Interface Mode selection. Default value: 0 (0: 4-wire interface; 1: 3-wire interface).
IF_INC	Register address automatically incremented during a multiple byte access with a serial interface (I ² C or SPI). Default value: 1 (0: disabled; 1: enabled)
BLE	Big/Little Endian Data selection. Default value 0 (0: data LSB @ lower address; 1: data MSB @ lower address)
SW_RESET	Software reset. Default value: 0 (0: normal mode; 1: reset device) This bit is cleared by hardware after next flash boot.

1. Boot request is executed as soon as internal oscillator is turned on. It is possible to set bit while in power-down mode, in this case it will be served at the next normal mode or sleep mode.

REGLAGE = 0b 0000 0110

- BLE = 1 pour inverser les octets des valeurs brutes

Les IMU de STMicroelectronic proposent par défaut la première adresse correspondant au bit de poids faible et la seconde de poids fort.

Dans les IMU TDK InvenSens c'est l'inverse et comme j'utilise la même routine arduino pour extraire les données brutes et que seul le LM6DS3 possède ce réglage du BLE je l'applique ici.

REGISTRE CTRL4_C (13h) Control register 4

Table 54. CTRL4_C register

XL_BW_SCAL_ODR	SLEEP_G	INT2_on_INT1	FIFO_TEMP_EN	DRDY_MASK	I2C_disable	MODE3_EN	STOP_ON_FTH
----------------	---------	--------------	--------------	-----------	-------------	----------	-------------

Table 55. CTRL4_C register description

XL_BW_SCAL_ODR	Accelerometer bandwidth selection. Default value: 0 (0 ⁽¹⁾ : bandwidth determined by ODR selection, refer to Table 48; 1 ⁽²⁾ : bandwidth determined by setting BW_XL[1:0] in CTRL1_XL (10h) register.)
SLEEP_G	Gyroscope sleep mode enable. Default value: 0 (0: disabled; 1: enabled)
INT2_on_INT1	All interrupt signals available on INT1 pad enable. Default value: 0 (0: interrupt signals divided between INT1 and INT2 pads; 1: all interrupt signals in logic or on INT1 pad)
FIFO_TEMP_EN	Enable temperature data as 4 th FIFO data set ⁽³⁾ . Default: 0 (0: disable temperature data as 4 th FIFO data set; 1: enable temperature data as 4 th FIFO data set)
DRDY_MASK	Configuration 1 ⁽⁴⁾ data available enable bit. Default value: 0 (0: DA timer disabled; 1: DA timer enabled)
I2C_disable	Disable I ² C interface. Default value: 0 (0: both I ² C and SPI enabled; 1: I ² C disabled, SPI only)
MODE3_EN	Enable auxiliary SPI interface (Mode 3, refer to Table 2). Default value: 0 (0: auxiliary SPI disabled; 1: auxiliary SPI enabled ⁽⁵⁾)
STOP_ON_FTH	Enable FIFO threshold level use. Default value: 0. (0: FIFO depth is not limited; 1: FIFO depth is limited to threshold level)

1. Filter used in high-performance mode only with ODR less than 3.33 kHz.

2. Filter used in high-performance mode only.

3. This bit is effective if the TIMER_PEDO_FIFO_EN bit of FIFO_CTRL2 register is set to 0.

4. In configuration 1, switching to combo mode, data are collected in FIFO only when both accelerometer and gyroscope are set. Switching to accelerometer only, data are collected in FIFO after filter setting.

5. Conditioned pads are: SDx, SCx, OCS

REGLAGE = 0b 0000 0000

- Laissé par défaut mais possibilité d'activer / désactiver le « sleep mode » du gyroscope et l'I²C.

REGISTRE CTRL5_C (14h) Control register 5

Table 56. CTRL5_C register

ROUNDING2	ROUNDING1	ROUNDING0	0 ⁽¹⁾	ST1_G	ST0_G	ST1_XL	ST0_XL
-----------	-----------	-----------	------------------	-------	-------	--------	--------

1. This bit must be set to '0' for the correct operation of the device

Table 57. CTRL5_C register description

ROUNDING[2:0]	Circular burst-mode (rounding) read from output registers. Default: 000 (000: no rounding; Others: refer to Table 58)
ST_G [1:0]	Angular rate sensor self-test enable. Default value: 00 (00: Self-test disabled; Other: refer to Table 59)
ST_XL [1:0]	Linear acceleration sensor self-test enable. Default value: 00 (00: Self-test disabled; Other: refer to Table 60)

Table 58. Output registers rounding pattern

ROUNDING[2:0]	Rounding pattern
000	No rounding
001	Accelerometer only
010	Gyroscope only
011	Gyroscope + accelerometer
100	Registers from SENSORHUB1_REG (2Eh) to SENSORHUB6_REG (33h) only
101	Accelerometer + registers from SENSORHUB1_REG (2Eh) to SENSORHUB6_REG (33h)
110	Gyroscope + accelerometer + registers from SENSORHUB1_REG (2Eh) to SENSORHUB6_REG (33h) and registers from SENSORHUB7_REG (34h) to SENSORHUB12_REG(39h)
111	Gyroscope + accelerometer + registers from SENSORHUB1_REG (2Eh) to SENSORHUB6_REG (33h)

Table 59. Angular rate sensor self-test mode selection

ST1_G	ST0_G	Self-test mode
0	0	Normal mode
0	1	Positive sign self-test
1	0	Not allowed
1	1	Negative sign self-test

Table 60. Linear acceleration sensor self-test mode selection

ST1_XL	ST0_XL	Self-test mode
0	0	Normal mode
0	1	Positive sign self-test
1	0	Negative sign self-test
1	1	Not allowed

REGLAGE = 0b 0000 0000 Laissé par défaut.

REGISTRE CTRL6_C (15h) Control register 6

Table 61. CTRL6_C register

TRIG_EN	LVLen	LVL2_EN	XL_HM_MODE	0 ⁽¹⁾	0 ⁽¹⁾	0 ⁽¹⁾	0 ⁽¹⁾
---------	-------	---------	------------	------------------	------------------	------------------	------------------

1. This bit must be set to '0' for the correct operation of the device.

Table 62. CTRL6_C register description

TRIG_EN	Gyroscope data edge-sensitive trigger enable. Default value: 0 (0: external trigger disabled; 1: external trigger enabled)
LVLen	Gyroscope data level-sensitive trigger enable. Default value: 0 (0: level-sensitive trigger disabled; 1: level sensitive trigger enabled)
LVL2_EN	Gyroscope level-sensitive latched enable. Default value: 0 (0: level-sensitive latched disabled; 1: level sensitive latched enabled)
XL_HM_MODE	High-performance operating mode disable for accelerometer ⁽¹⁾ . Default value: 0 (0: high-performance operating mode enabled; 1: high-performance operating mode disabled)

1. Normal and low-power mode depends on the ODR setting, for details refer to Table 47.

REGLAGE = 0b 0000 0000 Laissé par défaut.

REGISTRE CTRL7_G (16h) Angular rate sensor control register 7

Table 63. CTRL7_G register

G_HM_MODE	HP_G_EN	HPCF_G1	HPCF_G0	HP_G_RST	ROUNDING_STATUS	0 ⁽¹⁾	0 ⁽¹⁾
-----------	---------	---------	---------	----------	-----------------	------------------	------------------

1. This bit must be set to '0' for the correct operation of the device.

Table 64. CTRL7_G register description

G_HM_MODE	High-performance operating mode disable for gyroscope ⁽¹⁾ . Default: 0 (0: high-performance operating mode enabled; 1: high-performance operating mode disabled)
HP_G_EN	Gyroscope high-pass filter enable. Default value: 0 (0: HPF disabled; 1: HPF enabled)
HP_G_RST	Gyro digital HP filter reset. Default: 0 (0: gyro digital HP filter reset OFF; 1: gyro digital HP filter reset ON)
ROUNDING_STATUS	Source register rounding function enable on STATUS_REG (1Eh) , FUNC_SRC (53h) and WAKE_UP_SRC (1Bh) registers. Default value: 0 (0: disabled; 1: enabled)
HPCF_G[1:0]	Gyroscope high-pass filter cutoff frequency selection. Default value: 00. Refer to Table 65 .

1. Normal and low-power mode depends on the ODR setting, for details refer to [Table 51](#).

REGLAGE = 0b 0000 0000 Laissé par défaut.

REGISTRE CTRL8_XL (17h) Linear acceleration sensor control register 8

Table 66. CTRL8_XL register

LPF2_XL_EN	HPCF_XL1	HPCF_XL0	0 ⁽¹⁾	0 ⁽¹⁾	HP_SLOPE_X_L_EN	0 ⁽¹⁾	LOW_PASS_ON_6D
------------	----------	----------	------------------	------------------	-----------------	------------------	----------------

1. This bit must be set to '0' for the correct operation of the device.

Table 67. CTRL8_XL register description

LPF2_XL_EN	Accelerometer low-pass filter LPF2 selection. Refer to Figure 5 .
HPCF_XL[1:0]	Accelerometer slope filter and high-pass filter configuration and cutoff setting. Refer to Table 68 . It is also used to select the cutoff frequency of the LPF2 filter, as shown in Table 69 . This low-pass filter can also be used in the 6D/4D functionality by setting the LOW_PASS_ON_6D bit of CTRL8_XL (17h) to 1.
HP_SLOPE_XL_EN	Accelerometer slope filter / high-pass filter selection. Refer to Figure 5 .
LOW_PASS_ON_6D	Low-pass filter on 6D function selection. Refer to Figure 5 .

Table 68. Accelerometer slope and high-pass filter selection and cutoff frequency

HPCF_XL[1:0]	Applied filter	HP filter cutoff frequency [Hz]
00	Slope	ODR_XL/4
01	High-pass	ODR_XL/100
10	High-pass	ODR_XL/9
11	High-pass	ODR_XL/400

Table 69. Accelerometer LPF2 cutoff frequency

HPCF_XL[1:0]	LPF2 digital filter cutoff frequency [Hz]
00	ODR_XL/50
01	ODR_XL/100
10	ODR_XL/9
11	ODR_XL/400

REGLAGE = 0b 0000 0000 Laissé par défaut.

REGISTRE : CTRL9_XL (18h) Linear acceleration sensor control register 9

Table 70. CTRL9_XL register

0 ⁽¹⁾	0 ⁽¹⁾	Zen_XL	Yen_XL	Xen_XL	SOFT_EN	0 ⁽¹⁾	0 ⁽¹⁾
------------------	------------------	--------	--------	--------	---------	------------------	------------------

1. This bit must be set to '0' for the correct operation of the device.

Table 71. CTRL9_XL register description

Zen_XL	Accelerometer Z-axis output enable. Default value: 1 (0: Z-axis output disabled; 1: Z-axis output enabled)
Yen_XL	Accelerometer Y-axis output enable. Default value: 1 (0: Y-axis output disabled; 1: Y-axis output enabled)
Xen_XL	Accelerometer X-axis output enable. Default value: 1 (0: X-axis output disabled; 1: X-axis output enabled)
SOFT_EN	Enable soft-iron correction algorithm for magnetometer ⁽¹⁾ . Default value: 0 (0: soft-iron correction algorithm disabled; 1: soft-iron correction algorithm disabled)

1. This bit is effective if the IRON_EN bit of [MASTER_CONFIG \(1Ah\)](#) is set to 1.

REGLAGE = 0b 0011 1000 Laissé par défaut.

REGISTRE : CTRL10_C (19h) Control register 10

Table 72. CTRL10_C register

0 ⁽¹⁾	0 ⁽¹⁾	Zen_G	Yen_G	Xen_G	FUNC_EN	PEDO_RST_STEP	SIGN_MOTION_EN
------------------	------------------	-------	-------	-------	---------	---------------	----------------

1. This bit must be set to '0' for the correct operation of the device.

Table 73. CTRL10_C register description

Zen_G	Gyroscope yaw axis (Z) output enable. Default value: 1 (0: Z-axis output disabled; 1: Z-axis output enabled)
Yen_G	Gyroscope roll axis (Y) output enable. Default value: 1 (0: Y-axis output disabled; 1: Y axis output enabled)
Xen_G	Gyroscope pitch axis (X) output enable. Default value: 1 (0: X-axis output disabled; 1: X-axis output enabled)
FUNC_EN	Enable embedded functionalities (pedometer, tilt, significant motion, sensor hub and ironing) and accelerometer HP and LPF2 filters (refer to Figure 5). Default value: 0 (0: disable functionalities of embedded functions and accelerometer filters; 1: enable functionalities of embedded functions and accelerometer filters)
PEDO_RST_STEP	Reset pedometer step counter. Default value: 0 (0: disabled; 1: enabled)
SIGN_MOTION_EN	Enable significant motion function. Default value: 0 (0: disabled; 1: enabled)

REGLAGE = 0b 0011 1000 Laissé par défaut.

18 ANNEXE 2 : DÉTAILS DES CALCULS AVEC L'ACCELEROMÈTRE

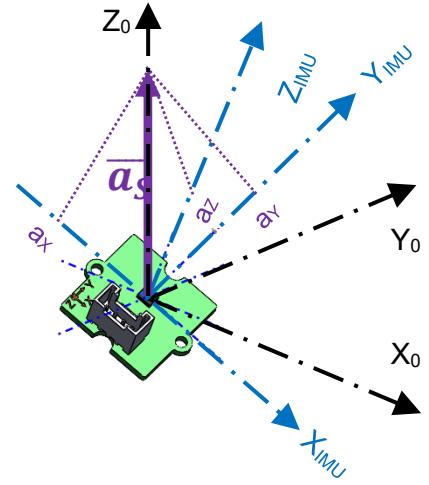
L'accéléromètre fournit les 3 composantes de l'accélération suivant les 3 axes du capteur dans le repère $(\mathbf{0}, \overrightarrow{\mathbf{X}_{IMU}}, \overrightarrow{\mathbf{Y}_{IMU}}, \overrightarrow{\mathbf{Z}_{IMU}})$:

$$\overrightarrow{a_{IMU}} = \begin{pmatrix} a_X \\ a_Y \\ a_Z \end{pmatrix} \quad \text{ou} \quad \overrightarrow{a_{IMU}} = a_X \cdot \overrightarrow{\mathbf{X}_{IMU}} + a_Y \cdot \overrightarrow{\mathbf{Y}_{IMU}} + a_Z \cdot \overrightarrow{\mathbf{Z}_{IMU}}$$

EN STATIQUE, et de par son architecture, le capteur mesure une accélération $\overrightarrow{a_S}$, image de l'accélération gravitationnelle donc verticale mais de composante **positive** (*voir le problème de l'accélération verticale*).

La mesure de l'accélération de pesanteur peut se définir dans le repère $(\mathbf{0}, \overrightarrow{\mathbf{X}_0}, \overrightarrow{\mathbf{Y}_0}, \overrightarrow{\mathbf{Z}_0})$:

$$\overrightarrow{a_S} = \begin{pmatrix} 0 \\ 0 \\ +\|\vec{g}\| \end{pmatrix} \quad \text{ou} \quad \overrightarrow{a_S} = \|\vec{g}\| \cdot \overrightarrow{\mathbf{Z}_0}$$



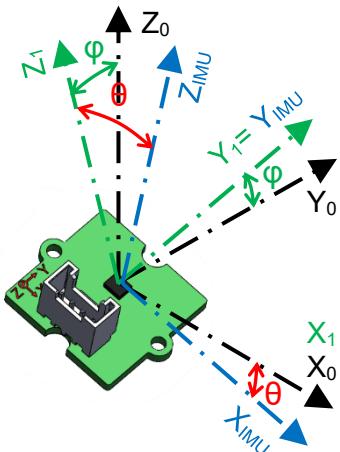
Lorsque le capteur est parfaitement horizontal, seule la composante a_Z mesure **l'opposée de la gravité** ($-\|\vec{g}\|$), les composantes a_X et a_Y sont alors nulles.

Comme expliqué au chapitre sur le paramétrage des angles de roulis et tangage, il existe 3 choix possibles pour définir ces angles :

CHOIX N°1

Première rotation de ROULIS puis celle de TANGAGE

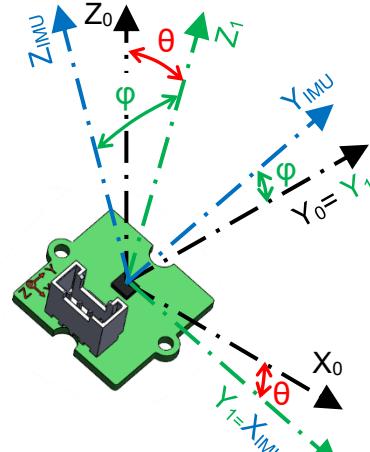
- ✓ La rotation de **ROULIS** d'un angle φ autour de l'axe $\overrightarrow{\mathbf{X}_0}$ auquel on associe un repère $(\mathbf{0}, \overrightarrow{\mathbf{X}_1}, \overrightarrow{\mathbf{Y}_1}, \overrightarrow{\mathbf{Z}_1})$
- ✓ La rotation de **TANGAGE** d'un angle θ autour de l'axe $\overrightarrow{\mathbf{Y}_1}$ qui permet de définir le repère de la centrale $(\mathbf{0}, \overrightarrow{\mathbf{X}_{IMU}}, \overrightarrow{\mathbf{Y}_{IMU}}, \overrightarrow{\mathbf{Z}_{IMU}})$



CHOIX N°2

Première rotation de **TANGAGE** puis celle de **ROULIS**

- ✓ La rotation de **TANGAGE** d'un angle θ autour de l'axe $\overrightarrow{\mathbf{Y}_0}$ auquel on associe un repère $(\mathbf{0}, \overrightarrow{\mathbf{X}_1}, \overrightarrow{\mathbf{Y}_1}, \overrightarrow{\mathbf{Z}_1})$
- ✓ La rotation de **ROULIS** d'un angle φ autour de l'axe $\overrightarrow{\mathbf{X}_1}$ qui permet de définir le repère de la centrale $(\mathbf{0}, \overrightarrow{\mathbf{X}_{IMU}}, \overrightarrow{\mathbf{Y}_{IMU}}, \overrightarrow{\mathbf{Z}_{IMU}})$

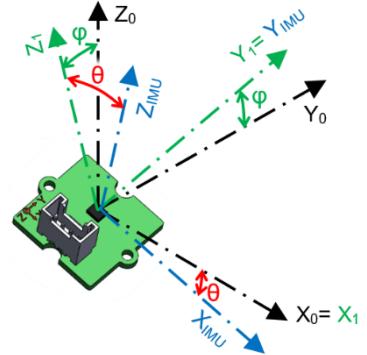
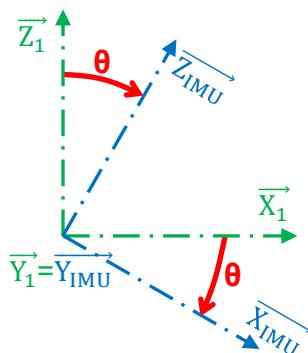
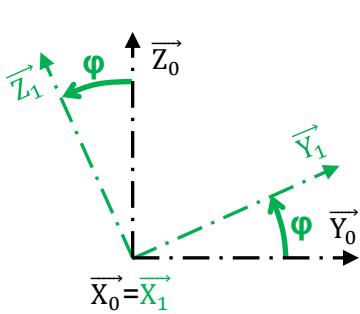


CHOIX N°3 :

Angles indépendants
Les angles sont définis par rapport aux plans

18.1 Choix N°1 : rotation de ROULIS puis de TANGAGE

18.1.1 Choix N°1 : Les changements de repères



Passage du repère $(O, \vec{X}_{IMU}, \vec{Y}_{IMU}, \vec{Z}_{IMU})$ au repère $(O, \vec{X}_0, \vec{Y}_0, \vec{Z}_0)$

$$\vec{X}_1 = \vec{X}_0$$

$$\vec{Y}_1 = \cos \varphi \cdot \vec{Y}_0 + \sin \varphi \cdot \vec{Z}_0$$

$$\vec{Z}_1 = -\sin \varphi \cdot \vec{Y}_0 + \cos \varphi \cdot \vec{Z}_0$$

$$\vec{X}_{IMU} = \cos \theta \cdot \vec{X}_1 - \sin \theta \cdot \vec{Z}_1$$

$$\vec{Y}_{IMU} = \vec{Y}_1$$

$$\vec{Z}_{IMU} = \sin \theta \cdot \vec{X}_1 + \cos \theta \cdot \vec{Z}_1$$

$$\vec{X}_{IMU} = \cos \theta \cdot \vec{X}_0 + \sin \theta \sin \varphi \cdot \vec{Y}_0 - \sin \theta \cos \varphi \cdot \vec{Z}_0$$

$$\vec{Y}_{IMU} = \cos \varphi \cdot \vec{Y}_0 + \sin \varphi \cdot \vec{Z}_0$$

$$\vec{Z}_{IMU} = \sin \theta \cdot \vec{X}_0 - \cos \theta \sin \varphi \cdot \vec{Y}_0 + \cos \theta \cos \varphi \cdot \vec{Z}_0$$

Passage du repère $(O, \vec{X}_0, \vec{Y}_0, \vec{Z}_0)$ au repère $(O, \vec{X}_{IMU}, \vec{Y}_{IMU}, \vec{Z}_{IMU})$

$$\vec{X}_0 = \vec{X}_1$$

$$\vec{Y}_0 = \cos \varphi \cdot \vec{Y}_1 - \sin \varphi \cdot \vec{Z}_1$$

$$\vec{Z}_0 = \sin \varphi \cdot \vec{Y}_1 + \cos \varphi \cdot \vec{Z}_1$$

$$\vec{X}_0 = \cos \theta \cdot \vec{X}_{IMU} - \sin \theta \cdot \vec{Z}_{IMU}$$

$$\vec{Y}_0 = \sin \varphi \sin \theta \cdot \vec{X}_{IMU} + \cos \varphi \cdot \vec{Y}_{IMU} - \sin \varphi \cos \theta \cdot \vec{Z}_{IMU}$$

$$\vec{Z}_0 = -\cos \varphi \sin \theta \cdot \vec{X}_{IMU} + \sin \varphi \cdot \vec{Y}_{IMU} + \cos \varphi \cos \theta \cdot \vec{Z}_{IMU}$$

$$\vec{X}_1 = \cos \theta \cdot \vec{X}_{IMU} + \sin \theta \cdot \vec{Z}_{IMU}$$

$$\vec{Y}_1 = \vec{Y}_{IMU}$$

$$\vec{Z}_1 = -\sin \theta \cdot \vec{X}_{IMU} + \cos \theta \cdot \vec{Z}_{IMU}$$

18.1.2 Choix N°1 : Composantes des vecteurs dans les différents repères

	\vec{a}_{IMU}
Dans R_{IMU}	a_x a_y a_z
Dans R_1	$a_x \cdot \cos \theta + a_z \cdot \sin \theta$ a_y $-a_x \cdot \sin \theta + a_z \cdot \cos \theta$
Dans R_0	$a_x \cdot \cos \theta + a_z \cdot \sin \theta$ $a_x \cdot \sin \theta \sin \varphi + a_y \cdot \cos \varphi - a_z \cdot \cos \theta \sin \varphi$ $-a_x \cdot \sin \theta \cos \varphi + a_y \cdot \sin \varphi + a_z \cdot \cos \theta \cos \varphi$

	\vec{a}_s
	$-\ \vec{g}\ \cdot \cos \varphi \sin \theta$
	$\ \vec{g}\ \cdot \sin \varphi$
	$\ \vec{g}\ \cdot \cos \varphi \cos \theta$
	0
	$\ \vec{g}\ \cdot \sin \varphi$
	$\ \vec{g}\ \cdot \cos \varphi$
	0
	0
	$+\ \vec{g}\ $

Habituellement, les calculs présentés ci-dessous utilisent des **matrices de rotations**, appelées aussi matrices de passage ou matrices de changement de base.

L'outil Mathématiques « matrice » n'étant pas au programme de Sciences de l'ingénieur ou STI2D en pré bac, les matrices de rotations ne sont, volontairement, pas utilisées ici.

Les démonstrations ci-dessous, sont donc limitées à l'utilisation d'expressions algébriques « simples » qui pourront être comprises par un élève de Terminale.

Pour en découvrir davantage sur les changements de repères :

[Repérage d'un solide par rapport à un repère - CPGE PTSI Réunion](#)

[ROBOTIQUE INDUSTRIELLE - Fabio MORBIDI - UPJV, Département EEA](#)

[ROBOTIQUE - Master Ingénierie et Technologie : Robotique - Université Louis Pasteur de Strasbourg](#)

18.1.3 CHOIX N°1 : Relations $\vec{a}_{IMU} = \vec{a}_S$

$$(1) \quad a_x = -\|\vec{g}\| \cdot \cos \varphi \sin \theta$$

$$(2) \quad a_y = \|\vec{g}\| \cdot \sin \varphi$$

$$(3) \quad a_z = \|\vec{g}\| \cdot \cos \varphi \cos \theta$$

$$(4) \quad a_x \cdot \cos \theta + a_z \cdot \sin \theta = 0$$

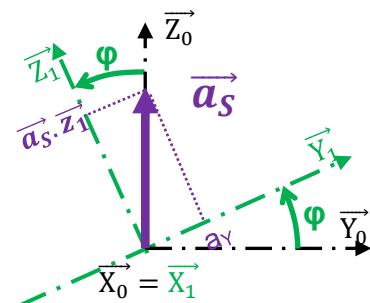
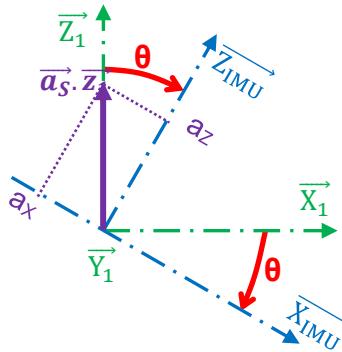
$$(5) \quad = (2)$$

$$(6) \quad -a_x \cdot \sin \theta + a_z \cdot \cos \theta = \|\vec{g}\| \cdot \cos \varphi$$

$$(7) \quad = (4)$$

$$(8) \quad a_x \cdot \sin \theta \sin \varphi + a_y \cdot \cos \varphi - a_z \cdot \cos \theta \sin \varphi = 0$$

$$(9) \quad -a_x \cdot \sin \theta \cos \varphi + a_y \cdot \sin \varphi + a_z \cdot \cos \theta \cos \varphi = \|\vec{g}\|$$



18.1.4 CHOIX N°1 : Relations pour le calcul du ROULIS

$$(4) \quad a_y = \|\vec{g}\| \cdot \sin \varphi$$

$$\frac{(2)}{(3)} \quad \frac{a_y}{a_z} = \frac{\|\vec{g}\| \cdot \sin \varphi}{\|\vec{g}\| \cdot \cos \varphi \cos \theta}$$

$$\frac{\sin \varphi}{\cos \varphi} = \frac{a_y}{a_z \cos \theta} = \frac{a_y}{\vec{a}_s \cdot \vec{z}_1} = \frac{a_y}{\pm \sqrt{a_x^2 + a_z^2}}$$

$$\frac{(2)}{(1)} \quad \frac{a_y}{a_x} = \frac{\|\vec{g}\| \cdot \sin \varphi}{-\|\vec{g}\| \cdot \cos \varphi \sin \theta}$$

$$\frac{\sin \varphi}{\cos \varphi} = \frac{a_y}{-\frac{a_x}{\sin \theta}} = \frac{a_y}{\vec{a}_s \cdot \vec{z}_1} = \frac{a_y}{\pm \sqrt{a_x^2 + a_z^2}}$$

$$\sin \varphi = \frac{a_y}{\|\vec{g}\|}$$

$$\tan \varphi = \frac{a_y}{\pm \sqrt{a_x^2 + a_z^2}}$$

Remarque : la composante de l'accélération sur l'axe \vec{z}_1 peut se calculer avec les composantes a_x et a_z : $\sqrt{a_x^2 + a_z^2}$ mais cette formule ne permet pas de connaître le sens de l'accélération !

Il faut donc prévoir de récupérer l'information en utilisant par exemple le signe de la composante a_z .

18.1.5 CHOIX N°1 : Relations pour le calcul du TANGAGE

$$(4) \quad a_x \cdot \cos \theta + a_z \cdot \sin \theta = 0$$

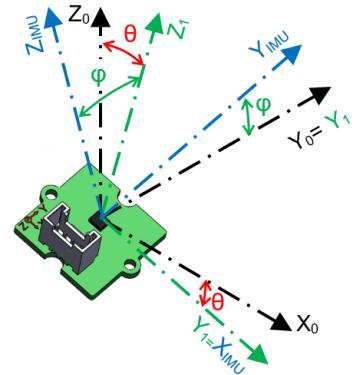
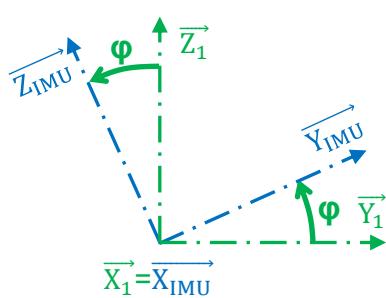
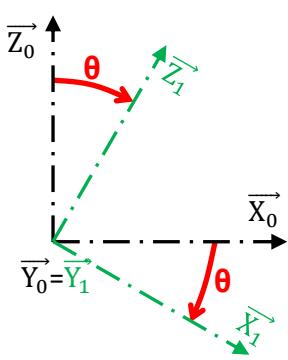
$$a_z \cdot \sin \theta = -a_x \cdot \cos \theta$$

$$\frac{\sin \theta}{\cos \theta} = -\frac{a_x}{a_z}$$

$$\tan \theta = -\frac{a_x}{a_z}$$

18.2 Choix N°2 : rotation de TANGAGE puis de ROULIS

18.2.1 Choix N°2 : Les changements de repères



Passage du repère $(0, \overrightarrow{X_{IMU}}, \overrightarrow{Y_{IMU}}, \overrightarrow{Z_{IMU}})$ au repère $(0, \overrightarrow{X_0}, \overrightarrow{Y_0}, \overrightarrow{Z_0})$

$$\overrightarrow{X_1} = \cos \theta \cdot \overrightarrow{X_0} - \sin \theta \cdot \overrightarrow{Z_0}$$

$$\overrightarrow{Y_1} = \overrightarrow{Y_0}$$

$$\overrightarrow{Z_1} = \sin \theta \cdot \overrightarrow{X_0} + \cos \theta \cdot \overrightarrow{Z_0}$$

$$\overrightarrow{X_{IMU}} = \overrightarrow{X_1}$$

$$\overrightarrow{Y_{IMU}} = \cos \varphi \cdot \overrightarrow{Y_1} + \sin \varphi \cdot \overrightarrow{Z_1}$$

$$\overrightarrow{Z_{IMU}} = -\sin \varphi \cdot \overrightarrow{Y_1} + \cos \varphi \cdot \overrightarrow{Z_1}$$

$$\overrightarrow{X_{IMU}} = \cos \theta \cdot \overrightarrow{X_0} - \sin \theta \cdot \overrightarrow{Z_0}$$

$$\overrightarrow{Y_{IMU}} = \sin \varphi \sin \theta \cdot \overrightarrow{X_0} + \cos \varphi \cdot \overrightarrow{Y_0} + \sin \varphi \cos \theta \cdot \overrightarrow{Z_0}$$

$$\overrightarrow{Z_{IMU}} = +\cos \varphi \sin \theta \cdot \overrightarrow{X_0} - \sin \varphi \cdot \overrightarrow{Y_0} + \cos \varphi \cos \theta \cdot \overrightarrow{Z_0}$$

Passage du repère $(0, \overrightarrow{X_0}, \overrightarrow{Y_0}, \overrightarrow{Z_0})$ au repère $(0, \overrightarrow{X_{IMU}}, \overrightarrow{Y_{IMU}}, \overrightarrow{Z_{IMU}})$

$$\overrightarrow{X_0} = \cos \theta \cdot \overrightarrow{X_1} + \sin \theta \cdot \overrightarrow{Z_1}$$

$$\overrightarrow{Y_0} = \overrightarrow{Y_1}$$

$$\overrightarrow{Z_0} = -\sin \theta \cdot \overrightarrow{X_1} + \cos \theta \cdot \overrightarrow{Z_1}$$

$$\overrightarrow{X_1} = \overrightarrow{X_{IMU}}$$

$$\overrightarrow{Y_1} = \cos \varphi \cdot \overrightarrow{Y_{IMU}} - \sin \varphi \cdot \overrightarrow{Z_{IMU}}$$

$$\overrightarrow{Z_1} = \sin \varphi \cdot \overrightarrow{Y_{IMU}} + \cos \varphi \cdot \overrightarrow{Z_{IMU}}$$

$$\overrightarrow{X_0} = \cos \theta \cdot \overrightarrow{X_{IMU}} + \sin \theta \sin \varphi \cdot \overrightarrow{Y_{IMU}} + \sin \theta \cos \varphi \cdot \overrightarrow{Z_{IMU}}$$

$$\overrightarrow{Y_0} = \cos \varphi \cdot \overrightarrow{Y_{IMU}} - \sin \varphi \cdot \overrightarrow{Z_{IMU}}$$

$$\overrightarrow{Z_0} = -\sin \theta \cdot \overrightarrow{X_{IMU}} + \cos \theta \sin \varphi \cdot \overrightarrow{Y_{IMU}} + \cos \theta \cos \varphi \cdot \overrightarrow{Z_{IMU}}$$

18.2.2 Choix N°2 : Composantes des vecteurs dans les différents repères

$$\overrightarrow{a_{IMU}}$$

$$\begin{cases} \text{Dans } \\ \text{R}_{IMU} \end{cases} \begin{cases} a_x \\ a_y \\ a_z \end{cases}$$

$$\overrightarrow{a_s}$$

$$\begin{cases} -\|\vec{g}\| \cdot \sin \theta \\ \|\vec{g}\| \cdot \cos \theta \sin \varphi \\ \|\vec{g}\| \cdot \cos \theta \cos \varphi \end{cases}$$

$$\begin{cases} a_x \\ a_y \cdot \cos \varphi - a_z \cdot \sin \varphi \\ a_y \cdot \sin \varphi + a_z \cdot \cos \varphi \end{cases}$$

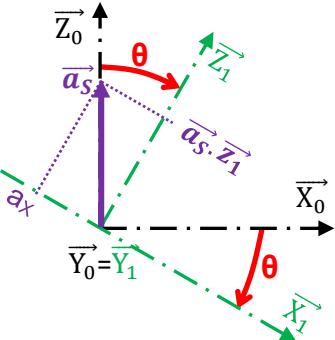
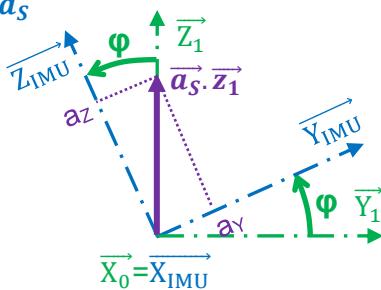
$$\begin{cases} -\|\vec{g}\| \cdot \sin \theta \\ 0 \\ \|\vec{g}\| \cdot \cos \theta \end{cases}$$

$$\begin{cases} \text{Dans } \\ \text{R}_0 \end{cases} \begin{cases} a_x \cdot \cos \theta + a_y \cdot \sin \varphi \sin \theta + a_z \cdot \cos \varphi \sin \theta \\ a_y \cdot \cos \varphi - a_z \cdot \sin \varphi \\ -a_x \cdot \sin \theta + a_y \cdot \sin \varphi \cos \theta + a_z \cdot \cos \varphi \cos \theta \end{cases}$$

$$\begin{cases} 0 \\ 0 \\ +\|\vec{g}\| \end{cases}$$

18.2.3 Choix N°2 : Relations $\vec{a}_{IMU} = \vec{a}_s$

- (1) $a_x = -\|\vec{g}\| \cdot \sin \theta$
- (2) $a_y = \|\vec{g}\| \cdot \cos \theta \sin \varphi$
- (3) $a_z = \|\vec{g}\| \cdot \cos \theta \cos \varphi$
- (4) = (1)
- (5) $a_y \cdot \cos \varphi - a_z \cdot \sin \varphi = 0$
- (6) $a_y \cdot \sin \varphi + a_z \cdot \cos \varphi = \|\vec{g}\| \cdot \cos \theta$
- (7) $a_x \cdot \cos \theta + a_y \cdot \sin \theta + a_z \cdot \cos \theta \sin \theta = 0$
- (8) = (5)
- (9) $-a_x \cdot \sin \theta + a_y \cdot \sin \theta \cos \theta + a_z \cdot \cos \theta \cos \theta = \|\vec{g}\|$



18.2.4 Choix N°2 : Relations pour le calcul du ROULIS

$$(5) \quad a_y \cdot \cos \varphi - a_z \cdot \sin \varphi = 0$$

$$\frac{\sin \varphi}{\cos \varphi} = \frac{a_y}{a_z}$$

$$\tan \varphi = \frac{a_y}{a_z}$$

$$(2) \quad \frac{a_y}{a_z} = \frac{\|\vec{g}\| \cdot \cos \theta \sin \varphi}{\|\vec{g}\| \cdot \cos \theta \cos \varphi}$$

$$(3) \quad \frac{a_y}{a_z} = \frac{-\|\vec{g}\| \cdot \sin \theta}{\|\vec{g}\| \cdot \cos \theta \cos \varphi}$$

18.2.5 Choix N°2 : Relations pour le calcul du TANGAGE

$$(1) \quad a_x = -\|\vec{g}\| \cdot \sin \theta$$

$$\sin \theta = -\frac{a_x}{\|\vec{g}\|}$$

$$\frac{(1)}{(2)} \quad \frac{a_x}{a_y} = \frac{-\|\vec{g}\| \cdot \sin \theta}{\|\vec{g}\| \cdot \cos \theta \sin \varphi}$$

$$\frac{\sin \theta}{\cos \theta} = \frac{a_x}{a_y} = \frac{a_x}{\vec{a}_s \cdot \vec{z}_1} = \frac{a_x}{\pm \sqrt{a_y^2 + a_z^2}}$$

$$\frac{(1)}{(3)} \quad \frac{a_x}{a_z} = \frac{-\|\vec{g}\| \cdot \sin \theta}{\|\vec{g}\| \cdot \cos \theta \cos \varphi}$$

$$\frac{\sin \theta}{\cos \theta} = \frac{a_x}{a_z} = \frac{a_x}{\vec{a}_s \cdot \vec{z}_1} = \frac{a_x}{\pm \sqrt{a_y^2 + a_z^2}}$$

$$\tan \theta = -\frac{a_x}{\pm \sqrt{a_y^2 + a_z^2}}$$

Remarque la composante de l'accélération sur l'axe \vec{z}_1 peut se calculer avec les composantes a_y et a_z : $\sqrt{a_y^2 + a_z^2}$ mais cette formule ne permet pas de connaître le sens de l'accélération !
 Il faut donc prévoir de récupérer l'information en utilisant par exemple le signe de la composante a_z .

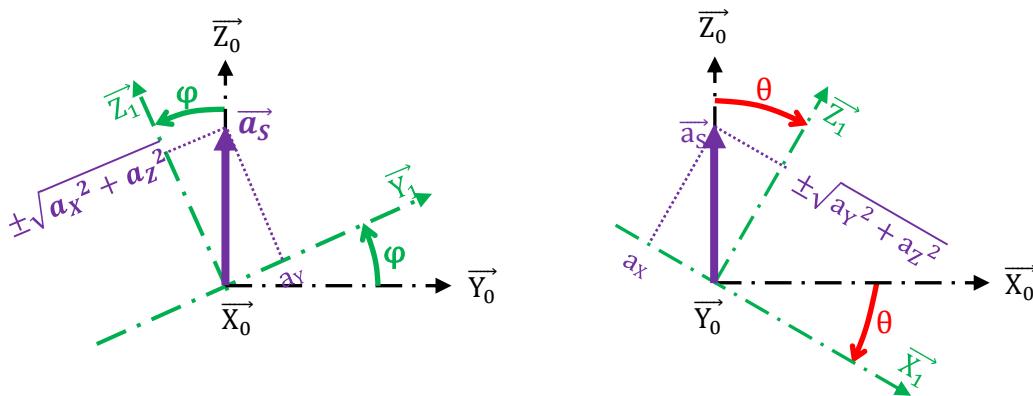
18.2.6 calcul des angles choix N°3 : TANGAGE ET ROULIS indépendants

Dans la bibliographie, on retrouve plusieurs notices d'application qui détaillent les calculs précédents et bien plus encore.

Certaines d'entre elles proposent une combinaison des calculs qui rendent les 2 rotations indépendantes.

Qui plus est, la définition des angles de LACET, ROULIS et TANGAGE telle que décrite [dans le paramétrage des angles de la centrale](#), n'est pas absolue et la bibliographie présente différentes descriptions dont certaines présentant les 3 angles totalement indépendants et paramétrés autours des axes fixes \vec{X}_0 , \vec{Y}_0 et \vec{Z}_0 .

Il est alors possible de représenter cette solution comme ci-dessous avec la rotation de LACET autour de l'axe \vec{Z}_0 , la rotation de ROULIS φ autour de l'axe \vec{X}_0 et la rotation de TANGAGE θ autour de l'axe \vec{Y}_0 .



$$\tan \varphi = \frac{a_y}{\pm \sqrt{a_x^2 + a_z^2}}$$

$$\tan \theta = \frac{a_x}{\pm \sqrt{a_y^2 + a_z^2}}$$

Remarque : le calcul $\sqrt{a_y^2 + a_z^2}$ ou $\sqrt{a_x^2 + a_z^2}$ ne permet pas de connaître le sens de l'accélération ! Il faut donc prévoir de récupérer l'information en utilisant par exemple le signe de la composante a_z .

Les notices d'application ci-dessous sont fournies par les constructeurs de centrales. On y retrouve des études plus poussées en particulier les limites de ces calculs :

[Application note 1057 - Analog Devices - Using an Accelerometer for Inclination Sensing](#)

[Application note 3182 - STMicroelectronics - Tilt measurement using a low-g 3-axis accelerometer](#)

[Application note 3046 - NXP Semiconductors - Tilt Sensing Using a Three-Axis](#)

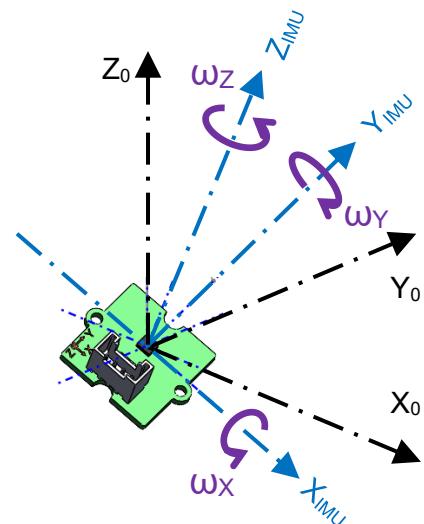
[Application note 5551 - STMicroelectronics - Precise and accurate tilt sensing in industrial applications](#)

19 ANNEXE 3 : DÉTAILS DES CALCULS AVEC LE GYROMÈTRE

Comme expliqué au chapitre du calcul des angles avec le gyromètre, la centrale inertie mesure le vecteur vitesse angulaire instantané $\overrightarrow{\Omega_{IMU}}$ dans le repère de la centrale ($0, \overrightarrow{X_{IMU}}, \overrightarrow{Y_{IMU}}, \overrightarrow{Z_{IMU}}$) :

$$\overrightarrow{\Omega_{IMU}} = \begin{pmatrix} \omega_X \\ \omega_Y \\ \omega_Z \end{pmatrix} \quad \text{ou} \quad \overrightarrow{\Omega_{IMU}} = \omega_X \cdot \overrightarrow{X_{IMU}} + \omega_Y \cdot \overrightarrow{Y_{IMU}} + \omega_Z \cdot \overrightarrow{Z_{IMU}}$$

Contrairement à l'accéléromètre, le gyromètre est capable de calculer l'angle de LACET mais il sera **relatif** par rapport à la position initiale.



Par contre les angles doivent être définis et donc calculés autour de leurs axes propres par intégration de leurs vitesses angulaires :

- ✓ ROULIS : $\varphi = \varphi_{\text{précédent}} + \varphi' \cdot \Delta t$
- ✓ TANGAGE : $\theta = \theta_{\text{précédent}} + \theta' \cdot \Delta t$
- ✓ LACET : $\psi = \psi_{\text{précédent}} + \psi' \cdot \Delta t$

φ' , θ' et ψ' , sont donc les vitesses angulaires respectives de ROULIS, TANGAGE et LACET qu'il convient de calculer autour de leurs axes respectifs.

Contrairement à l'accéléromètre qui est insensible à la rotation de LACET, il faut donc tenir compte, ici, de la rotation de LACET pour les calculs.

On se retrouve donc avec un paramétrage un peu plus complexe de 3 rotations successives avec 2 bases intermédiaires entre la base liée à la terre et celle liée à la centrale.

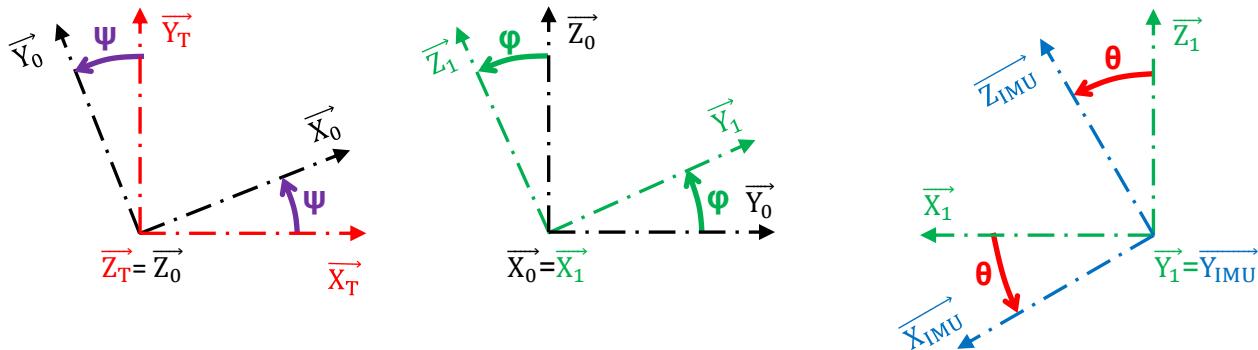
Comme le cas classique des angles d'Euler on se retrouve avec 12 possibilités et donc autant de choix possibles :

- ✓ LACET puis ROULIS puis TANGAGE
- ✓ LACET puis TANGAGE puis ROULIS
- ✓ ROULIS puis TANGAGE puis LACET
- ✓ TANGAGE puis ROULIS puis LACET
- ✓ ...

Il est difficile d'être exhaustif pour tous les cas précédents et les détails suivants sont limités au choix N°1 LACET, puis ROULIS puis TANGAGE.

19.1.1 Le paramétrage LACET puis ROULIS puis TANGAGE

Le paramétrage correspond à celui du choix N°1 du calcul des angles avec l'accéléromètre auquel on ajoute un angle supplémentaire ψ pour définir la base $(\mathbf{0}, \vec{X}_0, \vec{Y}_0, \vec{Z}_0)$ par rapport à une base Terrestre $(\mathbf{0}, \vec{X}_T, \vec{Y}_T, \vec{Z}_T)$ dont \vec{X}_T désignera le nord magnétique ou terrestre.



On peut définir le vecteur rotation de la base $\vec{\Omega}_{\text{IMU}}$ à la base $\vec{\Omega}_{\text{R}_T}$:

$$\vec{\Omega}_{\text{R}_{\text{IMU}} \rightarrow \text{R}_T} = \vec{\Omega}_{\text{R}_{\text{IMU}} \rightarrow \text{R}_1} + \vec{\Omega}_{\text{R}_1 \rightarrow \text{R}_0} + \vec{\Omega}_{\text{R}_0 \rightarrow \text{R}_T}$$

Avec $\vec{\Omega}_{\text{R}_{\text{IMU}} \rightarrow \text{R}_1} = \psi' \cdot \vec{Z}_T = \psi' \cdot \vec{Z}_0$, $\vec{\Omega}_{\text{R}_1 \rightarrow \text{R}_0} = \varphi' \cdot \vec{X}_0 = \varphi' \cdot \vec{X}_1$ et $\vec{\Omega}_{\text{R}_0 \rightarrow \text{R}_T} = \theta' \cdot \vec{Y}_T = \theta' \cdot \vec{Y}_{\text{IMU}}$

$$\vec{\Omega}_{\text{R}_{\text{IMU}} \rightarrow \text{R}_T} = \varphi' \cdot \vec{X}_1 + \theta' \cdot \vec{Y}_1 + \psi' \cdot \vec{Z}_0$$

Ainsi que le vecteur vitesse instantanée mesuré par le gyromètre :

$$\vec{\Omega}_{\text{IMU}} = \omega_X \cdot \vec{X}_{\text{IMU}} + \omega_Y \cdot \vec{Y}_{\text{IMU}} + \omega_Z \cdot \vec{Z}_{\text{IMU}}$$

Pour exprimer les relations des vitesses de ROULIS θ' , LACET φ' et TANGAGE ψ' , en fonction des composantes ω_X , ω_Y et ω_Z on peut définir l'égalité suivante :

$$\vec{\Omega}_{\text{R}_{\text{IMU}} \rightarrow \text{R}_T} = \vec{\Omega}_{\text{IMU}}$$

Mais il est nécessaire de l'exprimer dans la même base ici $(\mathbf{0}, \vec{X}_1, \vec{Y}_1, \vec{Z}_1)$ et, pour ce faire, exprimer les axes \vec{X}_{IMU} , \vec{Y}_{IMU} et \vec{Z}_{IMU} mais aussi l'axe \vec{Z}_0 dans cette même base :

$$\vec{X}_{\text{IMU}} = \cos \theta \cdot \vec{X}_1 - \sin \theta \cdot \vec{Z}_1$$

$$\vec{Z}_0 = \sin \varphi \cdot \vec{Y}_1 + \cos \varphi \cdot \vec{Z}_1$$

$$\vec{Y}_{\text{IMU}} = \vec{Y}_1$$

$$\vec{Z}_{\text{IMU}} = \sin \theta \cdot \vec{X}_1 + \cos \theta \cdot \vec{Z}_1$$

$$\vec{\Omega}_{\text{R}_{\text{IMU}} \rightarrow \text{R}_T} = \varphi' \cdot \vec{X}_1 + \theta' \cdot \vec{Y}_1 + \psi' \cdot (\sin \varphi \cdot \vec{Y}_1 + \cos \varphi \cdot \vec{Z}_1)$$

$$\vec{\Omega}_{\text{R}_{\text{IMU}} \rightarrow \text{R}_T} = \varphi' \cdot \vec{X}_1 + (\theta' + \psi' \cdot \sin \varphi) \cdot \vec{Y}_1 + \psi' \cdot \cos \varphi \cdot \vec{Z}_1$$

$$\vec{\Omega}_{\text{IMU}} = \omega_X \cdot \vec{X}_{\text{IMU}} + \omega_Y \cdot \vec{Y}_{\text{IMU}} + \omega_Z \cdot \vec{Z}_{\text{IMU}}$$

$$\vec{\Omega}_{\text{IMU}} = \omega_X \cdot (\cos \theta \cdot \vec{X}_1 - \sin \theta \cdot \vec{Z}_1) + \omega_Y \cdot \vec{Y}_1 + \omega_Z \cdot (\sin \theta \cdot \vec{X}_1 + \cos \theta \cdot \vec{Z}_1)$$

$$\vec{\Omega}_{\text{IMU}} = (\omega_X \cdot \cos \theta + \omega_Z \cdot \sin \theta) \cdot \vec{X}_1 + \omega_Y \cdot \vec{Y}_1 + (-\omega_X \cdot \sin \theta + \omega_Z \cdot \cos \theta) \cdot \vec{Z}_1$$

Et ainsi obtenir les relations suivantes :

- ✓ Direct pour le ROULIS φ' .

$$\varphi' = \omega_X \cdot \cos \theta + \omega_Z \cdot \sin \theta$$

- ✓ Direct pour le LACET ψ' :

$$\psi' \cdot \cos \varphi = -\omega_X \cdot \sin \theta + \omega_Z \cdot \cos \theta$$

$$\psi' = \frac{-\omega_X \cdot \sin \theta + \omega_Z \cdot \cos \theta}{\cos \varphi}$$

- ✓ Mais pour le TANGAGE θ' il faut « combiner » :

$$\theta' + \psi' \cdot \sin \varphi = \omega_Y$$

$$\theta' + \left(\frac{-\omega_X \cdot \sin \theta + \omega_Z \cdot \cos \theta}{\cos \varphi} \right) \cdot \sin \varphi = \omega_Y$$

$$\theta' = \omega_Y - \frac{-\omega_X \cdot \sin \theta + \omega_Z \cdot \cos \theta}{\cos \varphi} \cdot \sin \varphi$$

$$\theta' = \omega_Y - (-\omega_X \cdot \sin \theta + \omega_Z \cdot \cos \theta) \cdot \frac{\sin \varphi}{\cos \varphi}$$

$$\theta' = \omega_Y + (\omega_X \cdot \sin \theta - \omega_Z \cdot \cos \theta) \cdot \tan \varphi$$

STMicroelectronics propose un Conseil de conception qui présente des formules à partir d'une autre définition des rotations :
[Design tip DT0060 - STMicroelectronics - Exploiting the gyroscope to update tilt measurement and eCompass](#)

20 ANNEXE 4 : LA FONCTION NUMÉRIQUE ATAN2

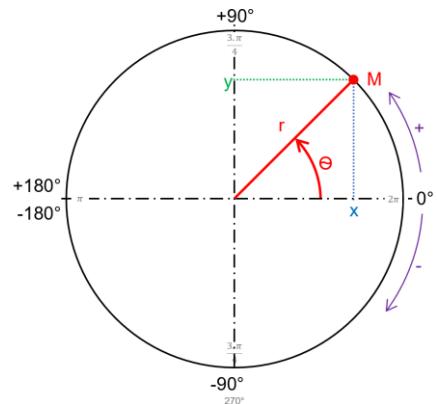
Atan2 est une fonction numérique, issue de la fonction Arctan qui a été spécialement définie pour le calcul d'un angle à partir de valeurs numériques de composantes vectorielles.

Comme la fonction Arctan, elle permet de calculer l'angle θ , de la coordonnée polaire du point M, en fonction de ses coordonnées cartésiennes.

$$\theta = \text{Arctan} \left(\frac{y}{x} \right) \quad \text{ou} \quad \theta = \text{Atan2}(y, x)$$

L'intérêt d'utiliser cette fonction à la place de l'Arctan :

- ✓ calculer l'angle dans l'intervalle $-180^\circ ; 180^\circ$ alors que l'arctan n'est valide que dans l'intervalle $-90^\circ ; 90^\circ$;
- ✓ renvoyer une valeur valide de $+180^\circ$ pour le cas $x = 0$. En effet $\text{Atan2}(1,0) = \text{Atan2}(-1,0) = +\pi$.



Il reste tout de même quelques inconvénients :

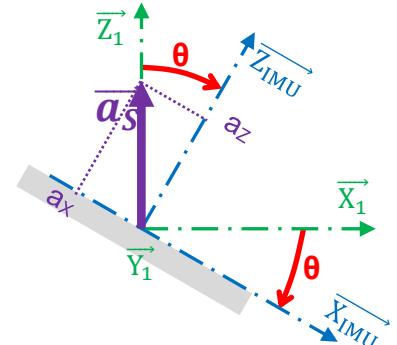
- ✓ la fonction Atan2 n'est pas définie pour le cas Atan2(0) mais le risque est faible puisque cela ne se produirait que dans le cas peu probable d'antigravité, avec le capteur parfaitement horizontal où les 3 composantes de l'accélération seraient nulle ;
- ✓ cette fonction n'empêche pas le problème du blocage de cardan qui présente des cas de non causalité pour des positions particulières.

L'utilisation de cette fonction est particulièrement adaptée pour le calcul des angles avec un accéléromètre.

Le schéma PLAN ci-contre représente un capteur qui s'inclinerait d'un angle θ autour d'un axe \vec{Y}_1 .

Le calcul de l'angle est donc fonction des composantes a_x et a_y :

$$\text{Soit } \theta = \text{Atan2}(a_z, a_x)$$

**LES CAS PARTICULIERS**

Θ	Rotation négative		Horizontale		Rotation positive	
	$\Theta = -180^\circ$	$\Theta = -90^\circ$	$\Theta = 0^\circ$	$\Theta = +90^\circ$	$\Theta = +180^\circ$	
Θ	-180°	-90°	0°	$+90^\circ$	$+180^\circ$	
a_x	0	-1	0	1	0	0
a_y	-1	0	1	0	0	-1
Θ_{calcul}	$\text{Atan2}(-1, 0) = +180^\circ$	$\text{Atan2}(0, -1) = -90^\circ$	$\text{Atan2}(1, 0) = 0^\circ$	$\text{Atan2}(0, 1) = +90^\circ$	$\text{Atan2}(-1, 0) = +180^\circ$	

21 ANNEXE 5 : LISTE DES FICHIERS ASSOCIES**21.1 Liste des programmes Arduino fournis pour les expérimentations :**

- ✓
- ✓
- ✓

21.2 Fichiers pour les bibliothèques

- ✓ Arduino_LSM6DS3-master.zip
- ✓ Grove_IMU_9DOF-master.zip
- ✓ MadgwickAHRS-master.zip Fichier pour installer la bibliothèque du filtre de Madgwick