

# **Realizzazione, taratura e verifica di un sensore inerziale per l'analisi del movimento**

Università degli studi di Roma “La Sapienza”

CDL Ingegneria delle Nanotecnologie

Corso: Laboratorio di micro/nano sensori



*Studente*  
**Stefano Gay**

*Professore*  
**Stefano Rossi**

# INDICE

<b>INTRODUZIONE .....</b>	<b>3</b>
LE INCERTEZZE DELLA MISURA .....	3
PRESENTAZIONE DELLA IMU.....	4
L'ACCELEROMETRO .....	4
IL GIROSCOPIO.....	8
<b>REALIZZAZIONE DELLA IMU .....</b>	<b>12</b>
ACQUISIZIONE DEI DATI.....	14
<b>LA CALIBRAZIONE DELL'ACCELEROMETRO .....</b>	<b>14</b>
REALIZZAZIONE DELLA TARATURA.....	16
RISULTATI .....	17
VERIFICA TARATURA ACCELEROMETRO .....	17
RISULTATI .....	18
<b>TARATURA DEL GIROSCOPIO .....</b>	<b>19</b>
REALIZZAZIONE DELLA TARATURA .....	21
I RISULTATI.....	22
VERIFICA DEL GIROSCOPIO .....	22
REALIZZAZIONE – Prima prova .....	22
REALIZZAZIONE – Seconda prova .....	24
<b>CALCOLO ANGOLI DI ROTAZIONE .....</b>	<b>24</b>
REALIZZAZIONE.....	26
Rotazione tra 0 e -90° lenta: .....	27
Rotazione tra 0° e +90° media: .....	30
Rotazione tra 0° e +90° veloce: .....	31
Rotazione tra -90° e +90° bassa: .....	33
Rotazione tra -90° e +90° media: .....	34
Rotazione tra -90° e +90° alta: .....	36
<b>CONCLUSIONI .....</b>	<b>37</b>
<b>LISTATI MATLAB .....</b>	<b>38</b>
TARATURA ACCELEROMETRO .....	38
VERIFICA TARATURA ACCELEROMETRO .....	39
TARATURA GIROSCOPIO .....	42
VERIFICA PARAMETRI DI TARATURA GIROSCOPIO .....	43
VERIFICA DERIVA GIROSCOPIO .....	44
CALCOLO ANGOLI DI ROTAZIONE .....	45
Main program .....	45
Funzione Calcola_angolo .....	46
Funzione calcola_versore.....	47
Funzione stampa_grafici .....	49
<b>BIBLIOGRAFIA.....</b>	<b>50</b>

## Introduzione

Il termine *navigazione* è solitamente utilizzato per indicare l'insieme dei metodi e delle tecniche che consentono di determinare la traiettoria di un oggetto in moto in un sistema di riferimento predeterminato. Allo scopo di determinare posizione, velocità e assetto di un corpo durante la navigazione vengono oggi utilizzati diversi sistemi, tra cui i più comuni sono GNSS (sistema di navigazione satellitare – Global Navigation Satellite Systems) e l'INS (o *navigazione inerziale* – *Inertial Navigation System*). Quest'ultimo è di interesse in questo lavoro: a partire dalla conoscenza del punto di partenza e dalle misure di accelerazione e di velocità angolare del corpo in movimento permette di risalire alla posizione del corpo e al percorso seguito.

In particolare, nella pratica comune si dispone all'interno di una piattaforma (IMU – *Inertial Measurement Unit*) una terna ortogonale di giroscopi ed un'analoga terna di accelerometri, affinché le velocità angolari e le accelerazioni lungo tutte e tre le direzioni del sistema di riferimento tridimensionale possano essere misurate. Il sistema può essere semplificato (perdendo ovviamente informazione) utilizzando una terna di accelerometri e un singolo giroscopio per l'analisi delle velocità angolare.

L'impiego di tali sensori nelle applicazioni civili, all'interno degli aeromobili di linea e dei veicoli rilevatori, è in gran parte dovuto allo sviluppo, nei primi anni Novanta, della tecnologia di costruzione delle fibre ottiche, che ha portato un miglioramento di circa un ordine di grandezza dell'accuratezza dei giroscopi ottici, e dei sensori elettromeccanici MEMS (acronimo di *Micro-Electro-Mechanical System*).

Sfruttando le eccellenti proprietà fisiche del silicio (più robusto, con caratteristiche termiche migliori e un peso specifico pari ad un terzo di quello dell'acciaio) è possibile infatti realizzare micro-sensori per la rilevazione di grandezze meccaniche che possono essere anche molto economici se prodotti su larga scala. A causa degli elevati errori sistematici ed accidentali, i sensori inerziali di tipo MEMS erano destinati inizialmente ad applicazioni di tipo *mass market*, caratterizzate in genere da scarse precisioni. Tuttavia, solo nell'ultimo decennio le accuratezze ottenibili con sensori di questo tipo hanno avuto un miglioramento di 3-4 ordini di grandezza, grazie agli studi sugli effetti della geometria della struttura e della sua dimensione, come pure a quelli sull'elettronica e l'assemblaggio.

Lo sviluppo negli ultimi vent'anni di sensori di questo tipo ha permesso di superare molti aspetti che in passato avevano precluso l'utilizzo dei sensori inerziali in diverse applicazioni, riducendone sensibilmente la dimensione, i costi ed i consumi, ed aprendo un nuovo mercato di sviluppo.

Un sistema di navigazione inerziale integra dunque le misure di accelerazione e di velocità angolare per fornire una soluzione di navigazione. Errori sistematici residui, anche di piccola entità, comportano quindi errori nella stima degli stati di navigazione che aumentano con il trascorrere del tempo. A causa di questi errori, i sistemi inerziali, nelle versioni diffuse ed accessibili anche per l'utenza civile, non consentono ad oggi di ottenere una soluzione di navigazione autonoma per lunghi periodi di tempo, e per questo devono essere associati a strumenti di natura diversa che "assistano" la navigazione.

## LE INCERTEZZE DELLA MISURA

Una unità di misura inerziale, come visto in precedenza, è composta da un insieme di sensori tali da consentire la stima di tutti gli stati della navigazione ad una elevata frequenza temporale. Trattandosi di sensori di natura elettro-meccanica, le misure effettuate risultano affette da errori di tipo sistematico (*bias* di misura, fattori di scala o non ortogonalità delle terne di sensori) ed accidentale (disturbi e rumori). I primi sono eliminabili attraverso procedure operative di calibrazione, che contribuiscono al miglioramento della conoscenza del sistema di misura. I secondi, invece, possono essere ridotti (ma quasi mai eliminati completamente) ripetendo più volte la misura, oppure sottoponendo i dati ad appositi algoritmi di *filtraggio* del segnale.

Come visto nel paragrafo precedente, una grossa diffusione nell'impiego dei sensori inerziali è legato allo sviluppo dei sensori MEMS, caratterizzati da una sensibile riduzione delle dimensioni, del costo e del peso. Tuttavia, il segnale in uscita da tali sensori è fortemente disturbato dalla presenza di un rumore ad altissima frequenza, che rende molto complesso il trattamento delle osservazioni grezze. In aggiunta, tali sensori sono nella maggior parte dei casi soggetti anche ad una incertezza *run-to-run*, in base alla quale alcuni errori di natura sistematica possono variare ad ogni accensione dello strumento, con un'incertezza che può essere fino a 10 volte maggiore rispetto all'entità del rumore di misura.

Per sfruttare appieno le potenzialità di tali tipologie di sensori, è pertanto necessario sottoporre gli stessi a diverse procedure, con lo scopo di valutare e modellare i diversi tipi di errori e il loro andamento nel tempo. A tale fine, nei paragrafi successivi vengono descritti i metodi di calibrazione e di filtraggio impiegati per la compensazione di tali errori dei sensori da noi utilizzati.

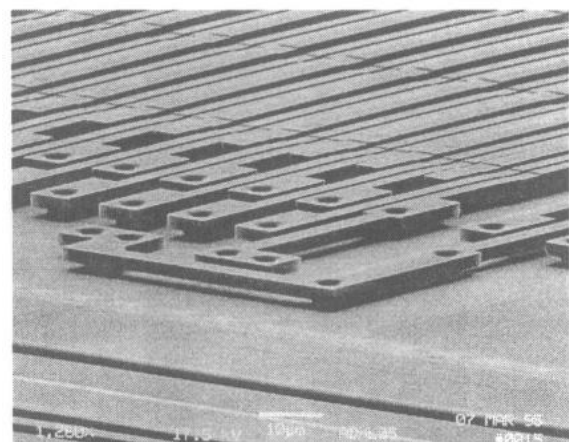
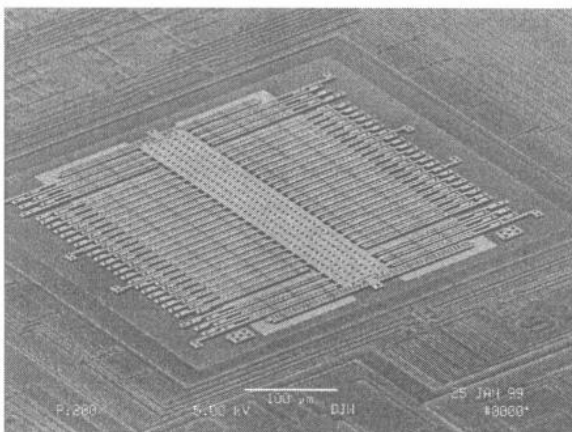
## PRESENTAZIONE DELLA IMU

La IMU di cui trattiamo è stata realizzata durante l'AA 2009-2010 durante il corso di "Laboratorio di micro/nano sensori" sotto la guida del Professor Stefano Rossi. Essa consiste in una scheda millefori su cui sono collegati un accelerometro triassiale e un giroscopio monoassiale, che ci permettono di rilevare tutte le informazioni necessarie relative alle accelerazioni e a quelle relative all'assetto del corpo. I componenti sono entrambi realizzati in tecnologia MEMS (*Micro Electro-Mechanical Systems*).

## L'ACCELEROMETRO

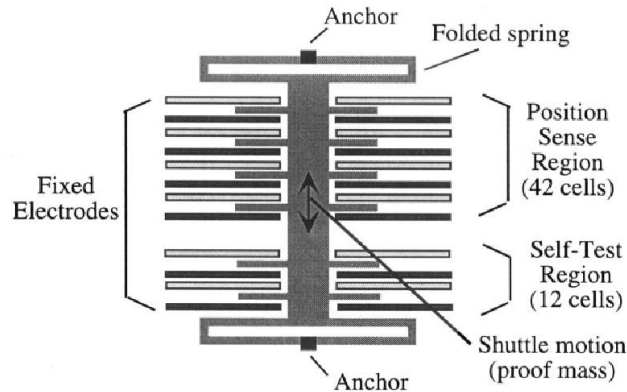
Nella maggior parte degli accelerometri, il principio di funzionamento è il medesimo: si basa sulla rilevazione dell'*inerzia* di una *massa* quando viene sottoposta ad una accelerazione.

La massa viene sospesa ad un *elemento elastico*, mentre un qualche tipo di sensore ne rileva lo spostamento rispetto alla struttura fissa del dispositivo. In presenza di un'accelerazione, la massa (che è dotata di una propria inerzia) si sposta dalla propria posizione di riposo in modo proporzionale all'accelerazione rilevata. Il sensore trasforma questo spostamento in un segnale elettrico acquisibile dai sistemi di misura. A riposo, con un asse sensibile rivolto verso il basso, sulla superficie terrestre un accelerometro misura circa 1 g, a causa dell'accelerazione gravitazionale data dalla massa della Terra. Di questa bisogna necessariamente tenere conto durante le misure. Esistono diversi tipi di accelerometri in commercio (piezoelettrici, piezoresistivi, per convezione di calore, ecc.) tra cui a livello commerciale stanno avendo un buon successo gli accelerometri MEMS di tipo capacitivo, realizzati in Silicio tramite microlavorazione. Un esempio di questo tipo di accelerometri possiamo vederlo nelle due figure qui sotto.

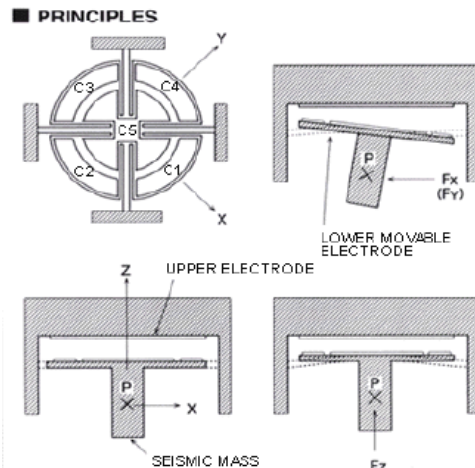


Al centro della figura sulla sinistra possiamo riconoscere la massa inerziale, sostenuta ai due estremi da due travi che costituiscono sostanzialmente la molla, ancorata alla scheda. I sensori capacitivi sono presenti sui lati, con una struttura a pettine identificata con il nome di *comb-drive*. Uno dei due denti è collegato a massa come riferimento. Al momento di una accelerazione lungo l'asse sensibile (in basso) la massa si

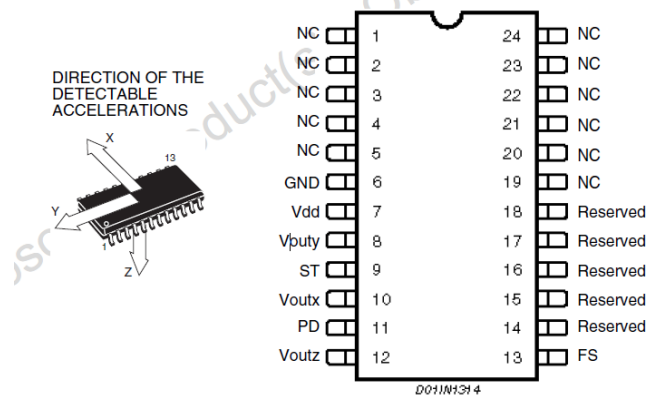
sposta, avvicinando o allontanando tra loro le facce degli elettrodi e variando così la capacità sulla linea. Questa è una delle strutture MEMS idealmente più semplici ed è oggi ampiamente utilizzata commercialmente per accelerometri *monoassiali*. Uno schema riassuntivo è presente in figura.



Entrando nel dettaglio del dispositivo utilizzato sulla nostra IMU, il dispositivo è un accelerometro *triassiale* schematizzato come in figura.



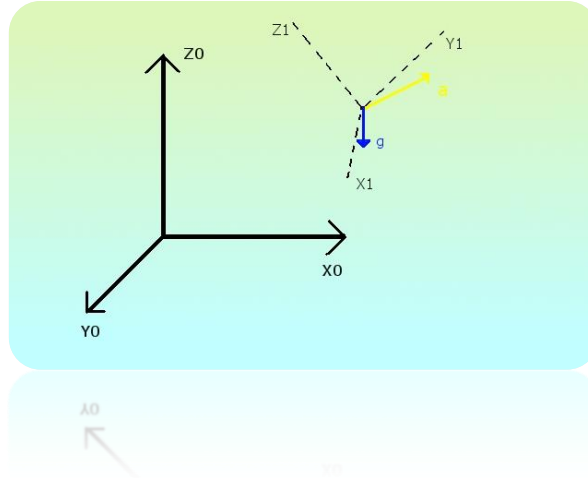
Una volta effettuato il packaging si presenta in questo modo:



I tre assi sensibili sono tutti perpendicolari tra loro, e l'asse z è rivolto verso il basso in modo da sentire  $-g$  in posizione stazionaria e perpendicolare al piano di terra. I collegamenti con il resto del circuito si effettuano semplicemente collegando (ad esempio tramite una saldatura a stagno) i vari cavi che portano i segnali necessari (compresi massa e alimentazione).

E' possibile utilizzare l'accelerometro con due modalità principali: misura dell'accelerazione inerziale (in caso  $a \gg g$ ) o dell'inclinazione del corpo (casi *quasi-statici*,  $a$  trascurabile rispetto a  $g$ ).

Nel primo caso avremo un'accelerazione  $a$  che sarà sempre da sommare al vettore di gravità  $g$ . In figura possiamo vedere il caso.



Con  $xyz_0$  indichiamo il riferimento assoluto, mentre  $xyz_1$  indica quello relativo al corpo. Il vettore di accelerazione inerziale  $\mathbf{a}$  viene sommato a  $\mathbf{g}$ . Stabilendo il verso di  $z_0$  come positivo  $\mathbf{g}$  risulterà negativo, portando quindi a un'accelerazione totale di  $\mathbf{a} - \mathbf{g}$ . Per un accelerometro monoassiale questo equivale a misurare

$$V = S \cdot \hat{\mathbf{n}}_1^T \cdot (\mathbf{a} - \mathbf{g})_1 + O$$

Dove  $S$  è la sensibilità dell'accelerometro,  $O$  il valore di "offset" ed  $\hat{\mathbf{n}}_1^T$  il versore che indica l'asse sensibile dell'accelerometro (nel riferimento inerziale). Passando al riferimento assoluto invece, dovremo considerare la rotazione effettuata per passare da un riferimento all'altro. Utilizzando le matrici di rotazione moltiplichiamo il vettore di accelerazione inerziale per la matrice di rotazione  ${}^1\mathbf{R}_0$  ovvero quella matrice che, moltiplicata, porta il sistema di riferimento assoluto a quello relativo.

$$V = S \cdot \hat{\mathbf{n}}_1^T \cdot {}^1\mathbf{R}_0 \cdot (\mathbf{a} - \mathbf{g})_0 + O$$

A questo punto possiamo vedere come cambiano le equazioni per un triassiale:

$$\begin{aligned} \begin{pmatrix} V_i \\ V_j \\ V_k \end{pmatrix}_{(3 \times 1)} &= \begin{pmatrix} S_i \cdot \hat{\mathbf{n}}_i^T \\ S_j \cdot \hat{\mathbf{n}}_j^T \\ S_k \cdot \hat{\mathbf{n}}_k^T \end{pmatrix}_{(3 \times 3)} \cdot {}^1\mathbf{R}_0_{(3 \times 3)} \cdot (\mathbf{a} - \mathbf{g})_{0(3 \times 1)} + \begin{pmatrix} O_i \\ O_j \\ O_k \end{pmatrix}_{(3 \times 1)} \\ &= \begin{bmatrix} S_i \cdot \hat{\mathbf{n}}_i^T & O_i \\ S_j \cdot \hat{\mathbf{n}}_j^T & O_j \\ S_k \cdot \hat{\mathbf{n}}_k^T & O_k \end{bmatrix}_{(3 \times 4)} \cdot \begin{bmatrix} {}^1\mathbf{R}_0 \cdot (\mathbf{a} - \mathbf{g})_0 \\ 1 \end{bmatrix}_{(4 \times 1)} \end{aligned}$$

Ovviamente il discorso può estendersi a un qualunque numero di accelerometri, per ognuno avremo un'equazione da aggiungere al sistema.

I versori  $\hat{\mathbf{n}}$  contengono le direzioni degli assi sensibili nel riferimento relativo, che, se sono orientati come  $xyz_0$ , corrispondono a quelli del riferimento assoluto:

$$\begin{aligned} n_i = n_x &= \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \\ n_j = n_y &= \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \\ n_k = n_z &= \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \end{aligned}$$

E questo ci permette di scrivere:

$$\begin{pmatrix} V_i \\ V_j \\ V_k \end{pmatrix}_{(3 \times 1)} = \begin{bmatrix} S_x & 0 & 0 & O_x \\ 0 & S_y & 0 & O_y \\ 0 & 0 & S_z & O_z \end{bmatrix} \cdot \begin{pmatrix} {}^1\alpha_x \\ {}^1\alpha_y \\ {}^1\alpha_z \\ 1 \end{pmatrix} = \begin{bmatrix} S_x \cdot {}^1\alpha_x + O_x \\ S_y \cdot {}^1\alpha_y + O_y \\ S_z \cdot {}^1\alpha_z + O_z \end{bmatrix}$$

Dove  ${}^1\alpha_x$ ,  ${}^1\alpha_y$  e  ${}^1\alpha_z$  sono rispettivamente le componenti dell'accelerazione totale (composizione dei vettori **a** e **g**) nel riferimento inerziale.

Nel caso *quasi-statico* l'accelerometro può essere utilizzato come inclinometro. Come detto il vettore di accelerazione inerziale deve essere trascurabile rispetto a quello di gravità **g** per essere valido questo procedimento. Conoscendo il valore dell'accelerazione a cui è sottoposto il sensore, possiamo risalire alla matrice di rotazione  ${}^1R_0$ . Per un singolo asse l'accelerometro è in grado di misurare l'accelerazione lungo il suo asse sensibile *s*, permettendo di conoscere l'angolo di inclinazione rispetto all'asse di riferimento, grazie alla conoscenza di **g**. Con questo sistema con un accelerometro triassiale è facile conoscere le rotazioni rispetto all'asse *x* e *y*, mentre risulta impossibile, pur utilizzando un più alto numero di accelerometri, conoscere la rotazione attorno a *z*: questo si comprende anche intuitivamente pensando al fatto che ruotando l'accelerometro perpendicolarmente rispetto all'accelerazione, la proiezione di questa sull'asse sensibile rimane comunque costante. Può essere dimostrato anche matematicamente: Applichiamo due rotazioni, una attorno all'asse *x* e una attorno a *y*. Questo corrisponde a una matrice di rotazione composta moltiplicando le due matrici di rotazione

$$\begin{aligned} {}^1R_0 &= \begin{pmatrix} \cos\varphi & 0 & -\sin\varphi \\ 0 & 1 & 0 \\ \sin\varphi & 0 & \cos\varphi \end{pmatrix} \\ {}^2R_1 &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{pmatrix} \\ {}^2R_0 &= {}^2R_1 {}^1R_0 = \begin{pmatrix} \cos\varphi & 0 & \sin\varphi \\ -\sin\varphi\sin\theta & \cos\theta & \cos\varphi\sin\theta \\ -\sin\varphi\cos\theta & -\sin\theta & \cos\varphi\cos\theta \end{pmatrix} \end{aligned}$$

E moltiplicando per questa il vettore **g**

$$\mathbf{g}_2 = {}^2R_0 \cdot \mathbf{g} = \begin{pmatrix} \cos\varphi & 0 & \sin\varphi \\ -\sin\varphi\sin\theta & \cos\theta & \cos\varphi\sin\theta \\ -\sin\varphi\cos\theta & -\sin\theta & \cos\varphi\cos\theta \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 0 \\ -g \end{pmatrix} = \begin{pmatrix} -g\sin\varphi \\ -g\cos\varphi\sin\theta \\ -g\cos\varphi\cos\theta \end{pmatrix}$$

Se i tre assi sensibili sono congruenti alla terna iniziale possiamo ottenere una misura di questo vettore:

$$\begin{pmatrix} V_x \\ V_y \\ V_z \end{pmatrix} = \begin{bmatrix} S_x & 0 & 0 & O_x \\ 0 & S_y & 0 & O_y \\ 0 & 0 & S_z & O_z \end{bmatrix} \cdot \begin{pmatrix} \mathbf{g} \\ 1 \end{pmatrix} = \begin{pmatrix} S_x \cdot (-g\sin\varphi) + O_x \\ S_y \cdot (-g\cos\varphi\sin\theta) + O_y \\ S_z \cdot (-g\cos\varphi\cos\theta) + O_z \end{pmatrix}$$

E infine ottenere gli angoli di rotazione:

$$\begin{aligned} \varphi &= \arcsin\left(-\frac{\alpha_x}{g}\right) = \arcsin\left(\frac{O_x - V_x}{g \cdot S_x}\right) \\ \theta &= \arctan\left(\frac{\alpha_y}{\alpha_z}\right) = \arctan\left(\frac{S_z(V_y - O_y)}{S_y(V_z - O_z)}\right) \end{aligned}$$

Dove  $\varphi$  e  $\theta$  sono rispettivamente gli angoli di rotazione attorno a *x* e *y*.

Vediamo ora cosa succede applicando una rotazione lungo l'asse *z*. Il vettore **g** diventa:

$$\mathbf{g}_3 = {}^3\mathbf{R}_0 \cdot \mathbf{g}_2 = \begin{pmatrix} \cos\gamma & \sin\gamma & 0 \\ -\sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} -g\sin\varphi \\ -g\cos\varphi\sin\theta \\ -g\cos\varphi\cos\theta \end{pmatrix} = \begin{pmatrix} -g\sin\varphi\cos\gamma - g\cos\varphi\sin\theta\sin\gamma \\ g\sin\varphi\sin\gamma - g\cos\varphi\sin\theta\cos\gamma \\ -g\cos\varphi\cos\theta \end{pmatrix}$$

Il sistema è però indeterminato, e non ci permette quindi di calcolare l'angolo  $\gamma$ .

Non sarà quindi possibile trovare i 6 gradi di libertà richiesti per la descrizione del moto del corpo con il solo accelerometro.

## IL GIROSCOPIO

Il *giroscopio* in generale è un dispositivo fisico rotante che, per effetto della *legge di conservazione del momento angolare*, tende a mantenere il suo asse di rotazione orientato in una direzione fissa.

Il *momento angolare totale* è la somma dei singoli momenti angolari che, di norma, non sono paralleli all'asse di rotazione: di conseguenza in generale anche il momento angolare totale non è parallelo all'asse di rotazione. Qualsiasi sia la forma del corpo è possibile trovare minimo tre direzioni fra di loro perpendicolari tali che, se il corpo ruota attorno a quell'asse, il momento angolare totale risulti parallelo all'asse di rotazione. Essi sono chiamati assi principali di inerzia ed i corrispondenti momenti di inerzia sono detti momenti principali di inerzia.

Gli assi principali di inerzia costituiscono un sistema di riferimento solidale col corpo ed, in generale, ruotano rispetto all'osservatore. Quando il corpo ha un qualche tipo di simmetria, gli assi principali di inerzia coincidono con questi assi di simmetria.

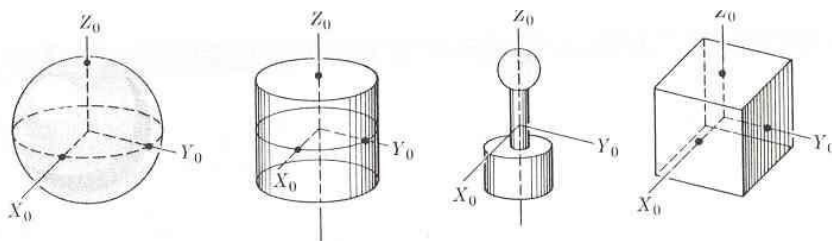


Fig. 10-5. Principal axis of symmetrical bodies.

Quando un corpo ruota attorno ad un asse principale di inerzia, il momento angolare totale è parallelo alla velocità angolare e si può scrivere la relazione vettoriale:

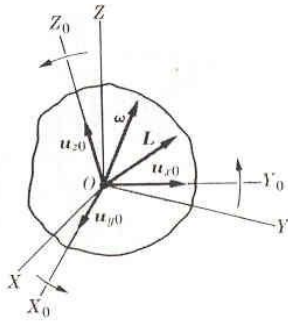
$$\mathbf{L} = \mathbf{I}\boldsymbol{\omega}$$

Nel caso più generale di rotazione del corpo rigido attorno ad un asse qualsiasi, il momento angolare totale può essere espresso rispetto agli assi principali di inerzia come:

$$\mathbf{L} = I_1 \omega_{x0} \mathbf{u}_{x0} + I_2 \omega_{y0} \mathbf{u}_{y0} + I_3 \omega_{z0} \mathbf{u}_{z0}$$

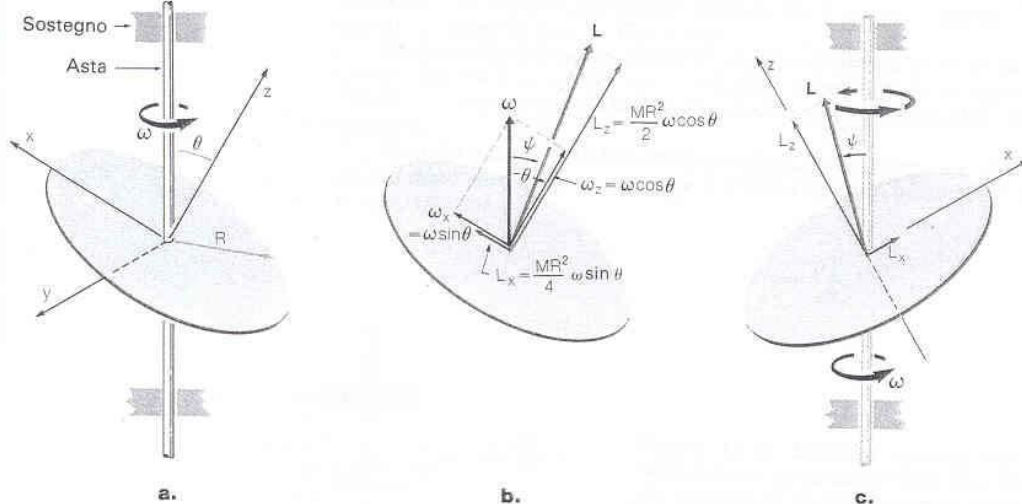
dove  $\mathbf{u}_{x0}$ ,  $\mathbf{u}_{y0}$ ,  $\mathbf{u}_{z0}$  sono i tre versori degli assi principali d'inerzia e  $\omega_{x0}$ ,  $\omega_{y0}$ ,  $\omega_{z0}$  sono le corrispondenti componenti della velocità angolare.  $\mathbf{L}$  ed  $\boldsymbol{\omega}$  hanno direzioni diverse, ma usando questa espressione,  $I_1$ ,  $I_2$ ,  $I_3$  sono quantità fisse (essendo riferite agli assi principali d'inerzia) che possono essere valutate per ogni corpo. I versori  $\mathbf{u}_{x0}$ ,  $\mathbf{u}_{y0}$ ,  $\mathbf{u}_{z0}$  non sono fissi nello spazio, ma ruotano con il corpo stesso, essendo ad esso solidali.





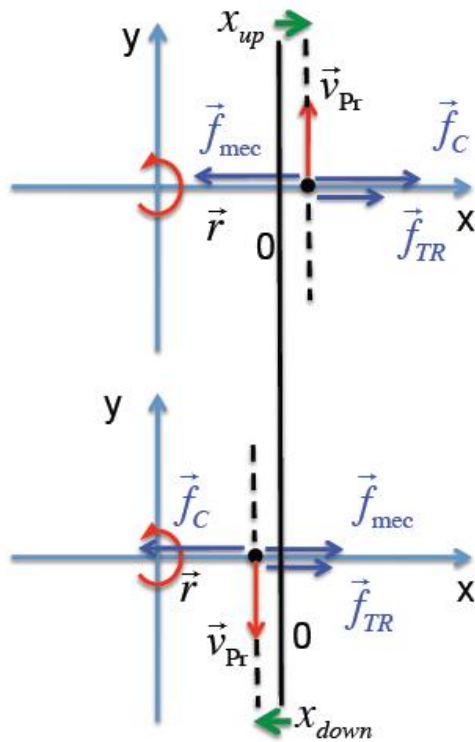
**Fig. 10-6.** Axes attached to the body and axes fixed in the laboratory.

Come esempio, possiamo vedere nel caso in figura come ‘nasca’ il vettore momento angolare: il corpo ruota attorno all’asse di rotazione con velocità angolare  $\omega$ , scomponibile rispetto alle tre direzioni  $\mathbf{u}_{x0}$ ,  $\mathbf{u}_{y0}$ ,  $\mathbf{u}_{z0}$  che caratterizzano gli assi principali d’inerzia. Il vettore momento angolare nasce dalla somma vettoriale delle tre componenti  $I_1 \omega_{x0} \mathbf{u}_{x0} + I_2 \omega_{y0} \mathbf{u}_{y0} + I_3 \omega_{z0} \mathbf{u}_{z0}$ , non coincide con il vettore  $\omega$ , e varia come direzione nello spazio durante la rotazione del corpo, non essendo i tre versori degli assi principali d’inerzia fissi nello spazio. Se il modulo del momento angolare rimane costante, si ha comunque una variazione del momento angolare, riconducibile all’azione di un momento della forza, legata alla variazione della direzione del momento angolare stesso.



**Figura 12-3** Un sottile disco piatto ruota attorno ad un’asta formante un angolo  $\theta$  con l’asse di simmetria. (a) Un sistema di assi Cartesiani  $x$ - $y$ - $z$  scelto lungo gli assi di simmetria del disco. Il sistema è solidale col disco e ruota con esso. (b) Le componenti del vettore velocità angolare  $\omega$  sui tre assi  $x$ ,  $y$  e  $z$ . Noti i momenti d’inerzia rispetto a questi assi, si determinano le corrispondenti componenti del momento angolare; la loro somma fornisce il vettore momento angolare istantaneo  $\mathbf{L}$ . (c) Il vettore momento angolare  $\mathbf{L}$  precessa attorno all’asse di rotazione. Il modulo di  $\mathbf{L}$  e l’angolo  $\psi$  tra  $\mathbf{L}$  e  $\omega$  si conservano.

Nella nostra esperienza si è scelto di utilizzare un giroscopio in tecnologia MEMS per gli stessi motivi per cui era stata fatta la stessa scelta nel caso dell’accelerometro. Il giroscopio MEMS, nel caso più semplice di giroscopio monoassiale, è costituito da una massa sospesa attraverso dei sistemi elastici che le permettono di muoversi in entrambe le direzioni  $x$  ed  $y$ . Un’elaborazione del segnale permette di ricavare la velocità angolare.

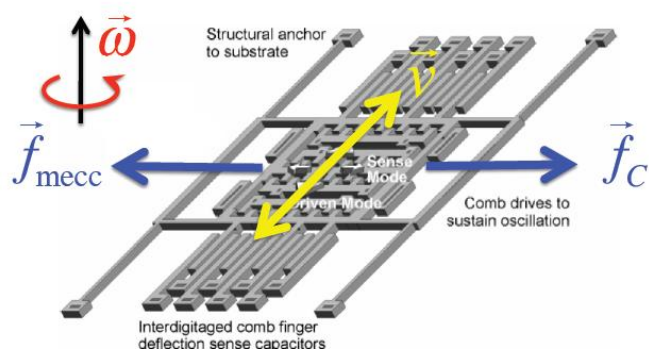
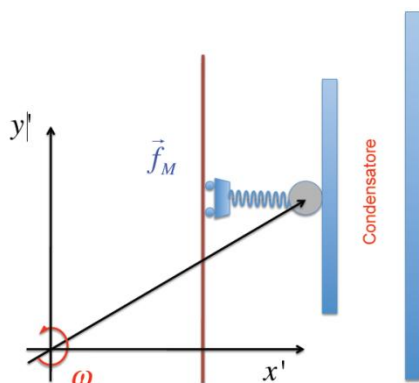


Idealmente il dispositivo funziona come schematizzato nella figura in alto: è presente una vibrazione (espressa da  $v_{pr}$  in figura), nel momento in cui una rotazione viene applicata, si sviluppa una forza di *coriolis*, ortogonale all'asse di rotazione e alla vibrazione principale, che sposta dal centro una massa, provocando una seconda vibrazione proporzionale alla rotazione applicata. In particolare la forza applicata segue la relazione:

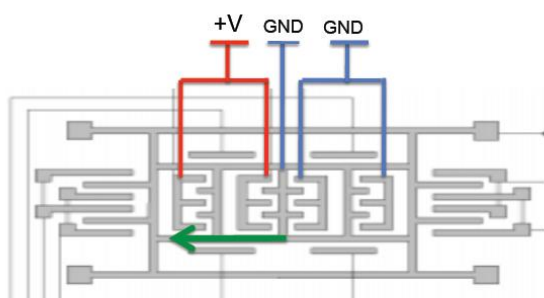
$$\vec{F} = -2m \cdot \vec{\omega} \times \vec{v}_{pr}$$

Con  $m$  massa del sensore,  $v_{pr}$  velocità di vibrazione e  $\omega$  velocità angolare agente.

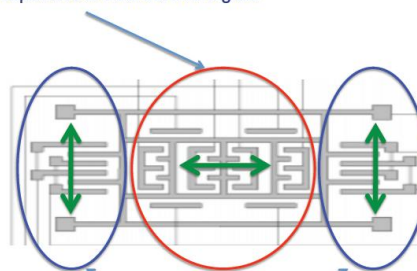
Un giroscopio vibrante può essere schematizzato come di seguito:



A seconda della rotazione applicata possono essere fatte avvicinare o allontanare due strutture all'interno del sensore. Questo suggerisce di usare due superfici parallele allo scopo di formare un condensatore: questo tipo di giroscopi MEMS è infatti di tipo capacitivo. La struttura è generalmente simile a quella riportata in figura in alto a destra. Anche questa, come per l'accelerometro ha una struttura di tipo *comb drive*, con le travi stesse usate come molle.

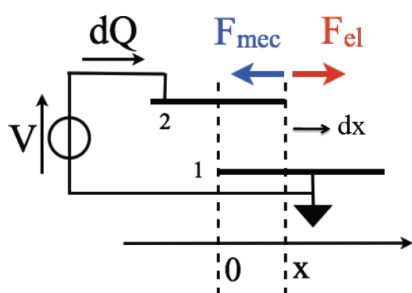


Attuatore capacitivo a elettrodi interdigitati



Sensore capacitivo a piatti paralleli

Applicando un potenziale il sensore entra in vibrazione (lungo un asse ad esempio x) e al momento della rotazione del corpo rispetto al secondo asse (z) avremo una vibrazione, e dunque una variazione di capacità all'interno del circuito di acquisizione, lungo il terzo asse (y).



Entrando nello specifico nella pratica è importante conoscere a cosa corrisponde il segnale rilevato dal sensore. Il giroscopio rileva in generale una tensione secondo l'equazione:

$$V = S \cdot (\hat{n}^T \omega) + O$$

Dove  $S$  è la sensibilità del giroscopio,  $O$  il valore di "offset" ed  $\hat{n}$  il versore che ne individua l'asse sensibile.  $\omega$  è il vettore delle rotazioni attorno ai tre assi  $x_0$ ,  $y_0$  e  $z_0$  del riferimento assoluto.

Definendo  $z$  l'asse sensibile di rotazione:

$$V = S \cdot (n_x \quad n_y \quad n_z) \cdot \begin{pmatrix} 0 \\ 0 \\ \omega_z \end{pmatrix} + O = S \cdot n_z \cdot \omega_z + O$$

Che si riduce se l'asse del giroscopio è allineato con l'asse  $z_0$  a

$$V = S \cdot \omega_z + O$$

Conoscendo il segnale misurato dal giroscopio è ovviamente possibile calcolare l'angolo integrando la velocità angolare:

$$\theta_z = \int_{t_1}^{t_2} \omega_z dt$$

Con  $t_1$  e  $t_2$  gli istanti di inizio e fine acquisizione. Sostituendo questa diventa:

$$\theta_z = \int_{t_1}^{t_2} \left( \frac{V - O}{S} \right) dt = \frac{1}{S} \int_{t_1}^{t_2} (V - O) dt$$

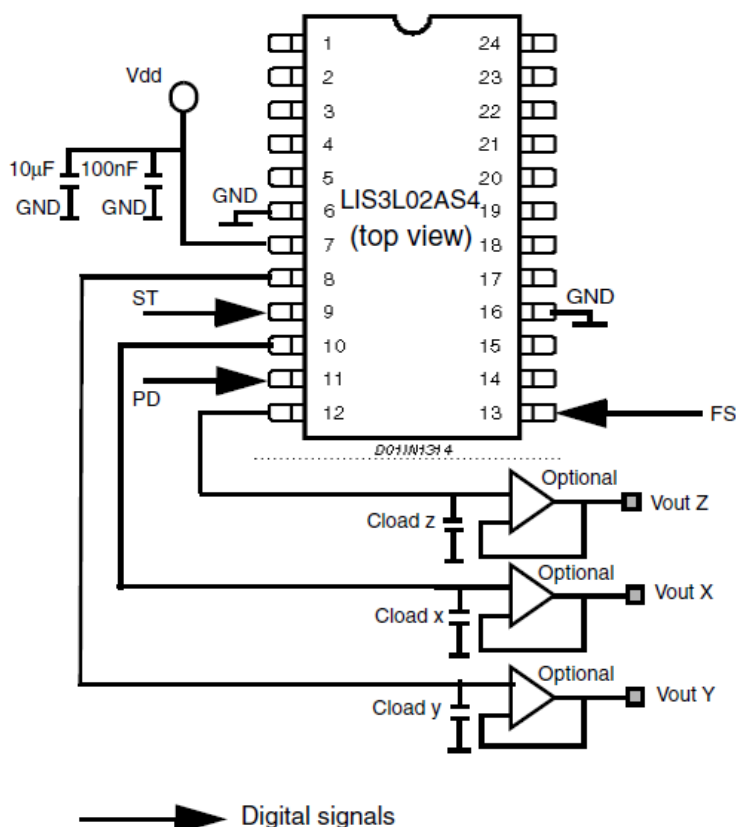
Da notare che questo procedimento, proprio a causa della presenza di una integrazione, è soggetto al fenomeno della deriva. La presenza di rumore o altre interferenze (ad esempio la stessa presenza della vibrazione primaria) comporta l'integrazione del rumore stesso. Il valore dell'integrale, e quindi dell'angolo

stimato, risulterà non corretto. È possibile applicare un filtro al fine di evitare parte di questo problema, che però non è completamente eliminabile in questo modo, e richiede ulteriori algoritmi di correzione, ad esempio l'uso di un accelerometro come inclinometro che corregga in parte l'errore, come vedremo in seguito.

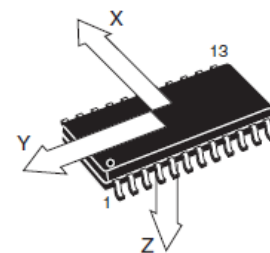
## REALIZZAZIONE DELLA IMU

Quello che intendiamo realizzare durante l'esperienza è la realizzazione di un sistema di navigazione che ci dia informazioni sul percorso seguito da un corpo e sulle rotazioni che subisce. Per realizzarlo abbiamo deciso di utilizzare un accelerometro triassiale ed un giroscopio monoassiale. L'asse di rotazione sensibile del giroscopio si è scelto lungo l'asse  $z_0$ : in questo modo mentre l'accelerometro ci darà informazioni sulle rotazioni intorno agli assi  $x$  e  $y$ , dal giroscopio avremo quelle relative all'asse  $z$ . Sarà inoltre possibile correggere, tramite opportuno algoritmo, almeno in parte, gli errori dovuti alla deriva del giroscopio, grazie ai dati dell'accelerometro.

Come accelerometro scegliamo l'Accelerometro triassiale LIS3L02AS4 della ST Microelectronics (package SMD, connettori SOIC a 24 pin). L'obiettivo è quello di montare questi componenti insieme su una basetta millefori con fori ramati, quindi abbiamo bisogno di un opportuno adattatore che faccia da supporto al sensore, e consenta di montarlo tramite pin. Il montaggio richiede la saldatura a stagno dei pin dell'accelerometro vero e proprio sull'adattatore, e poi dell'adattatore sulla scheda millefori. Tutte le saldature vengono eseguite a mano. In figura possiamo vedere lo schema circuitale utilizzato, presente sul datasheet dell'accelerometro stesso, e seguito da noi nel montaggio.



DIRECTION OF THE DETECTABLE ACCELERATIONS

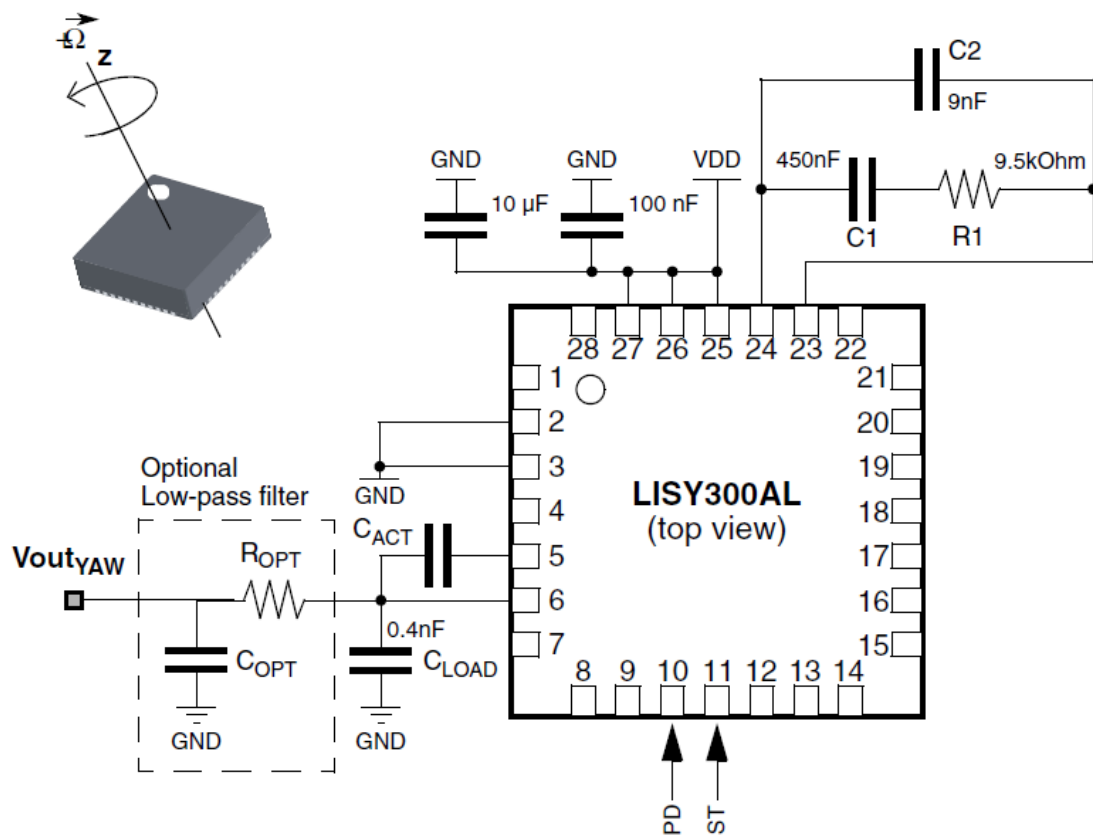


Verranno montati sulla scheda, come suggerito dal datasheet, 2 condensatori da 100nF e 10µF per stabilizzare la tensione in entrata  $V_{dd}$ , e 3 condensatori da 33nF per effettuare un filtro passa basso (taglio del rumore alle alte frequenze) sulle uscite.

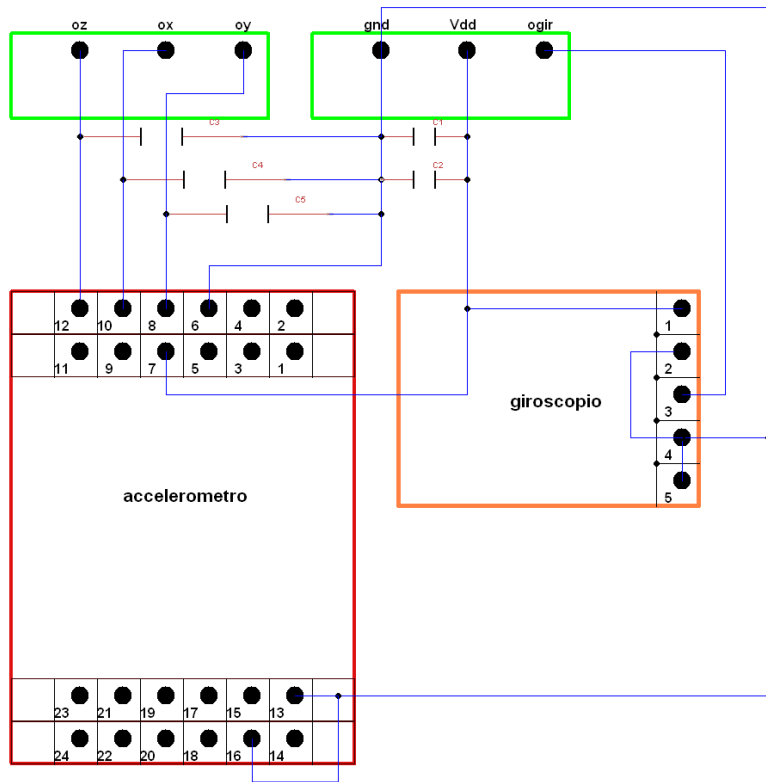
Il giroscopio è un LISY300AL della ST Microelectronics (package SMD, connettori LGA a 28 pin). Questo componente, come si vede in figura, non ha bisogno di adattatore, in quanto esso viene fornito dalla casa produttrice già montato pronto per il montaggio su millefori.



Lo schema circuitale è riportato in figura



Per gli ingressi e le uscite si utilizzano 2 connettori a tre porte. La IMU verrà interfacciata tramite la Ni-Elvis al calcolatore, dove i dati verranno raccolti attraverso la suite NI ELVIS e valutati in tempo reale tramite il software LabView. Il post processing verrà eseguito tramite il software Matlab. Il disegno della scheda è stato oggetto di studio, allo scopo di ridurre il più possibile l'ingombro occupato.



Ox oy e oz sono rispettivamente le uscite relative agli assi x,y e z dell'accelerometro, mentre ogir è l'uscita dell'asse del giroscopio.  $V_{dd}$  è la tensione di alimentazione, continua, e settata come da datasheet a 3,3V. Gnd è ovviamente la tensione di riferimento a massa. I condensatori hanno capacità rispettivamente:

- $C1=100\text{nF}$
- $C2=10\mu\text{F}$
- $C3=C4=C5=33\text{nF}$

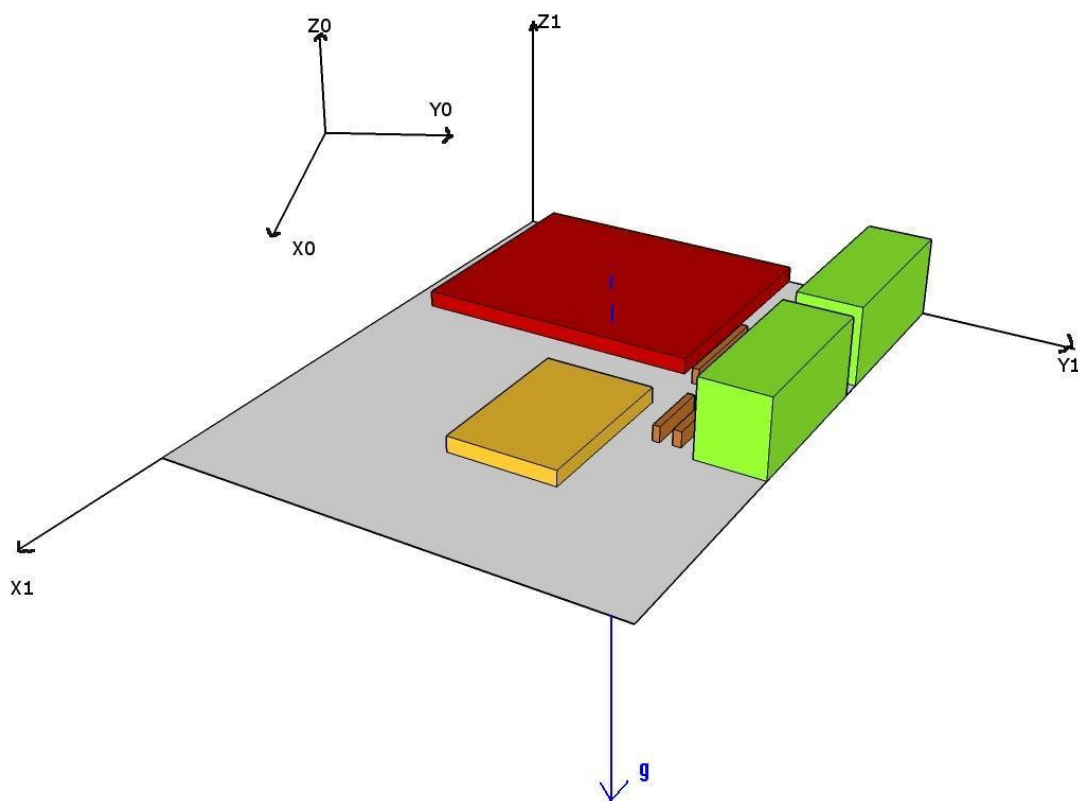
Le connessioni sono state effettuate collegando alla scheda millefori, tramite un punto di saldatura di stagno, un filo di rame per ogni pista. La richiesta di minimizzare lo spazio utilizzato ha portato a uno spazio occupato di soli 17X18 pin della millefori.

## ACQUISIZIONE DEI DATI

Per ottenere l'acquisizione dei dati a partire dall'IMU è stata utilizzata la suite della National Instrument NI Elvis. I dati acquisiti vengono poi processati tramite un software real time in linguaggio LabView. Il software, sviluppato sempre durante questa esperienza, è costituito schematicamente da un ciclo di acquisizione in loop che suddivide in 4 tracce, una per ogni porta collegata, il segnale proveniente dalla IMU. Le 4 tracce rappresentano  $V_g$ ,  $V_x$ ,  $V_y$ ,  $V_z$  (tensione del giroscopio, e tensione relativa ai 3 assi dell'accelerometro x y e z) e vengono salvate su file testuale in formato ASCII, in quattro colonne. Le colonne sono separate da una tabulazione. Sempre tramite LabView si imposta la tensione di alimentazione alla scheda (3.3V).

## LA CALIBRAZIONE dell'ACCELEROMETRO

Lo scopo principale della calibrazione è quello di determinare i principali sistematismi che incidono sulla misura, in maniera da poter prevedere in maniera sufficientemente accurata il comportamento del sensore inerziale e di conseguenza le accuratèzze raggiungibili durante l'utilizzo. Una volta determinati, questi errori possono essere parzialmente compensati in maniera da accrescere le prestazioni del sensore stesso. Per quanto concerne i sensori inerziali di medio-alto costo, tutte le procedure di calibrazione sono definite e normate dall'*Institute of Electrical and Electronics Engineers (IEEE)* con una serie di documenti pubblicati nel corso degli anni. Non volendo ricorrere a tali procedure di calibrazione, complesse e costose, è possibile ricorrere a test molto più semplici, che non richiedono strumentazioni od ambienti appositi ma che consentono comunque di verificare la conformità del sensore alle accuratèzze finali richieste, determinando i valori approssimati dei più importanti sistematismi. Questo gruppo di prove è costituito da *test statici* e *test dinamici*, che sfruttano la risposta che il sensore ha rispetto a grandezze fisiche note, come ad esempio l'**accelerazione gravitazionale**.



Sfruttando quest'ultima cercheremo di calcolare per primi i parametri di *Sensibilità* ( $S_x S_y S_z$ ) e di *offset* ( $O_x O_y O_z$ ), relativi all'accelerometro.

In generale, come abbiamo visto, per un sistema di riferimento come quello in figura vale:

$$\begin{pmatrix} V_x \\ V_y \\ V_z \end{pmatrix}_{(3 \times 1)} = \begin{bmatrix} S_x \cdot \hat{n}_x^T & O_x \\ S_y \cdot \hat{n}_y^T & O_y \\ S_z \cdot \hat{n}_z^T & O_z \end{bmatrix}_{(3 \times 4)} \cdot \begin{bmatrix} {}^1R_0 \cdot (a - g)_0 \\ 1 \end{bmatrix}_{(4 \times 1)}$$

Dove  $\hat{n}$  sono i versori relativi agli assi sensibili dell'accelerometro nel sistema di riferimento  $xyz_1$ . L'idea è quella di imporre rotazioni note e, conoscendo il valore dell'accelerazione totale, ricavare i parametri cercati. Per imporre l'accelerazione totale nota si sceglie di utilizzare l'accelerazione gravitazionale  $g$ . Questo ovviamente permetterà l'utilizzo corretto dello strumento (intervallo di linearità) solo in condizione

quasi-statica (campo di accelerazione da  $-\mathbf{g}$  a  $+\mathbf{g}$ ), e quindi come inclinometro. È da sottolineare che se si esce dalla condizione di quasi-staticità, quindi in caso vengano imposte all'accelerometro delle altre accelerazioni almeno comparabili con  $g$ , i risultati calcolati con questo setup non sono da considerarsi attendibili. Le rotazioni note verranno applicate matematicamente tramite una matrice di rotazione che porti dal sistema di riferimento assoluto a quello relativo. Questa sarà necessariamente una matrice composizione delle possibili rotazione attorno agli assi  $x$ ,  $y$  e  $z$  quindi del tipo:

$${}^1\mathbf{R}_0 = \mathbf{R}(x) \cdot \mathbf{R}(y) \cdot \mathbf{R}(z) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{pmatrix} \cdot \begin{pmatrix} \cos\beta & 0 & -\sin\beta \\ 0 & 1 & 0 \\ \sin\beta & 0 & \cos\beta \end{pmatrix} \cdot \begin{pmatrix} \cos\gamma & \sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Con  $\alpha$ ,  $\beta$  e  $\gamma$  angoli relativi alle rotazioni rispettivamente rispetto a  $x_0, y_0$  e  $z_0$ .

Per ogni asse imporremo  $g$  una volta positivo e una negativo arrivando a ottenere 6 misure.

$$\left[ \begin{pmatrix} V_x \\ V_y \\ V_z \end{pmatrix}_1 \dots \begin{pmatrix} V_x \\ V_y \\ V_z \end{pmatrix}_n \right]_{(3 \times n)} = \begin{bmatrix} S_x \cdot \hat{n}_x^T & O_x \\ S_y \cdot \hat{n}_y^T & O_y \\ S_z \cdot \hat{n}_z^T & O_z \end{bmatrix}_{(3 \times 4)} \cdot \left[ \begin{pmatrix} {}^1\mathbf{R}_0 \cdot (-\mathbf{g})_0 \\ 1 \end{pmatrix}_1 \dots \begin{pmatrix} {}^1\mathbf{R}_0 \cdot (-\mathbf{g})_0 \\ 1 \end{pmatrix}_n \right]_{(4 \times n)}$$

Per risolvere il sistema è necessario invertire il secondo membro del prodotto, ma questo ovviamente non è possibile non essendo la matrice quadrata. Per risolvere questo problema utilizziamo una funzione matematica che approssima l'operazione di inversione su una matrice rettangolare, la *pseudoinversa*, presente anche su Matlab con il nome *pinv*. Il sistema diventa quindi:

$$\begin{bmatrix} S_x \cdot \hat{n}_x^T & O_x \\ S_y \cdot \hat{n}_y^T & O_y \\ S_z \cdot \hat{n}_z^T & O_z \end{bmatrix}_{(3 \times 4)} = \left[ \begin{pmatrix} V_x \\ V_y \\ V_z \end{pmatrix}_1 \dots \begin{pmatrix} V_x \\ V_y \\ V_z \end{pmatrix}_n \right]_{(3 \times n)} \cdot \text{pinv} \left( \left[ \begin{pmatrix} {}^1\mathbf{R}_0 \cdot (-\mathbf{g})_0 \\ 1 \end{pmatrix}_1 \dots \begin{pmatrix} {}^1\mathbf{R}_0 \cdot (-\mathbf{g})_0 \\ 1 \end{pmatrix}_n \right]_{(4 \times n)} \right)$$

Dove la matrice a sinistra è costituita da

$$\begin{bmatrix} S_x \cdot \hat{n}_x^T & O_x \\ S_y \cdot \hat{n}_y^T & O_y \\ S_z \cdot \hat{n}_z^T & O_z \end{bmatrix} = \begin{bmatrix} S_x \cdot n_{x,1} & S_x \cdot n_{x,2} & S_x \cdot n_{x,3} & O_x \\ S_x \cdot n_{y,1} & S_x \cdot n_{y,2} & S_x \cdot n_{y,3} & O_y \\ S_x \cdot n_{z,1} & S_x \cdot n_{z,2} & S_x \cdot n_{z,3} & O_z \end{bmatrix}$$

Le prime tre colonne contengono i vettori  $S_i \cdot \hat{n}_i^T$ . Questi forniscono l'orientamento del singolo asse sensibile dell'accelerometro ed il loro modulo è pari alla sensibilità  $S_i$ . Nell'ultima colonna sono presenti gli offset relativi a ogni asse. Da questa matrice abbiamo tutti i dati che ci servono: la sensibilità del singolo asse (ad esempio il modulo di  $S_x \cdot \hat{n}_x^T$  per l'asse  $x$ ), il versore ( $S_x \cdot \hat{n}_x^T / S_x$ ) e l'offset ( $O_x$ ).

## Realizzazione della taratura

Si effettuano 6 misure (vengono qui riportate le rotazioni eseguite):

- (0,0,0)
- (180,0,0)
- (0,-90,0)
- (0,+90,0)
- (-90,0,0)
- (+90,0,0)



Tramite Matlab si esegue la risoluzione del sistema. Il programma Matlab viene realizzato come segue

- Caricamento file dati. I file devono essere privi di eventuali header e suddivisi in 4 colonne dati:

<i>segnale giroscopio</i>	<i>Segnale accelerometro asse x</i>	<i>Segnale accelerometro asse y</i>	<i>Segnale accelerometro asse z</i>
---------------------------	---	---	---

- Definizione parametri. Vengono definiti i parametri che verranno utilizzati nel resto del programma. Tra questi gli angoli e le matrici di rotazione relative alle prove fatte (vd sopra).
- Calcolo delle medie. Viene effettuata la media di ognuno dei tre segnali relativi ai tre assi dell'accelerometro. Questa è un'approssimazione che ci riduce notevolmente i calcoli, possibile in quanto ci troviamo in regime quasi statico e quindi le misure sono simili tra loro.
- Soluzione. Si moltiplica la matrice contenente i segnali mediati per la pseudoinversa della matrice contenente  $\begin{pmatrix} {}^1R_0 \cdot (-g)_0 \\ 1 \end{pmatrix}$ . Da questa si calcolano come visto S, gli offset e i versori. I risultati vengono stampati a video.

## Risultati

$$S_x = 0.0733 \frac{Vs^2}{m} \quad S_y = 0.0736 \frac{Vs^2}{m} \quad S_z = 0.0744 \frac{Vs^2}{m}$$

Essendo la sensibilità definita come il rapporto tra tensione e accelerazione, stimiamo che più o meno per tutti gli assi è possibile misurare un'accelerazione pari a  $0.14 \frac{m}{s^2}$  per ogni centesimo di volt.

Gli offset risultano:

$$O_x = 1.8996V \quad O_y = 1.8084V \quad O_z = 1.9336V$$

La tensione di offset è compresa tra 1.8 e 1.9 per i tre assi, il che non è perfettamente concorde da quanto riportato dal datasheet, dove i tre offset dovrebbero essere circa la metà della tensione di alimentazione (avendo alimentato la scheda a 3.3V ci aspettiamo circa 1.65V di offset); questo può essere dovuto a una tensione di alimentazione non perfetta (probabilmente più elevata) rispetto al valore dichiarato di 3.3V oppure qualche accoppiamento resistivo/capacitivo sulla IMU.

Infine i versori:

$$\begin{aligned} \text{versore asse x} &= 0.9997 \quad -0.0089 \quad 0.0244 \\ \text{versore asse y} &= 0.0171 \quad 0.9992 \quad 0.0367 \\ \text{versore asse z} &= -0.0095 \quad 0.0054 \quad -0.9999 \end{aligned}$$

Come avevamo visto dal datasheet il versore dell'asse z deve risultare invertito e in effetti così è.

## Verifica Taratura Accelerometro

Per verificare la taratura dell'accelerometro possiamo fare due tipi di controllo: il primo sulla tensione misurata, il secondo sull'accelerazione misurata. Il primo si effettua facilmente applicando la formula vista in precedenza:

$$\begin{bmatrix} \begin{pmatrix} V_x \\ V_y \\ V_z \end{pmatrix}_1 \dots \begin{pmatrix} V_x \\ V_y \\ V_z \end{pmatrix}_n \end{bmatrix}_{(3 \times n)} = \begin{bmatrix} S_x \cdot \hat{n}_x^T & O_x \\ S_y \cdot \hat{n}_y^T & O_y \\ S_z \cdot \hat{n}_z^T & O_z \end{bmatrix}_{(3 \times 4)} \cdot \begin{bmatrix} ({}^1R_0 \cdot (-g)_0)_1 \dots ({}^1R_0 \cdot (-g)_0)_n \end{bmatrix}_{(4 \times n)}$$

In questo modo otteniamo le tensioni per ogni singola prova, da confrontare con le tensioni misurate dall'accelerometro. Le rotazioni in questo caso sono note, così come i parametri  $S$ ,  $\hat{n}$  e  $O$ . Useremo delle rotazioni diverse da quelle con cui è stata effettuata la taratura, e leggermente più complesse da realizzare, il che potrebbe portare errori dovuti alla realizzazione pratica del test stesso.

Il secondo controllo sull'accelerazione si attua semplicemente applicando, per ogni misura, la matrice di rotazione relativa al vettore di gravità  $g$ , e confrontandolo con il vettore calcolato dalla tensione dell'accelerometro; quest'ultima si calcola applicando:

$$\begin{bmatrix} (g)_1 \dots (g)_{1n} \end{bmatrix}_{(4 \times n)} = pinv \left( \begin{bmatrix} S_x \cdot \hat{n}_x^T & O_x \\ S_y \cdot \hat{n}_y^T & O_y \\ S_z \cdot \hat{n}_z^T & O_z \end{bmatrix} \right)_{(4 \times 3)} \cdot \begin{bmatrix} \begin{pmatrix} V_x \\ V_y \\ V_z \end{pmatrix}_1 \dots \begin{pmatrix} V_x \\ V_y \\ V_z \end{pmatrix}_n \end{bmatrix}_{(3 \times n)}$$

Dal confronto riportiamo anche una stima dell'errore di taratura.

Le rotazioni testate:

1.  $-90^\circ, 0^\circ, 0^\circ$
2.  $90^\circ, 0^\circ, 0^\circ$
3.  $45^\circ, 0^\circ, 0^\circ$
4.  $-45^\circ, 0^\circ, 0^\circ$
5.  $0^\circ, -45^\circ, 0^\circ$
6.  $0^\circ, 45^\circ, 0^\circ$
7.  $0^\circ, -90^\circ, 0^\circ$
8.  $0^\circ, 90^\circ, 0^\circ$
9.  $180^\circ, 0^\circ, 0^\circ$

## Risultati

I risultati sono stati ottenuti tramite l'applicazione Matlab in appendice. Il programma è suddiviso in sezioni come segue:

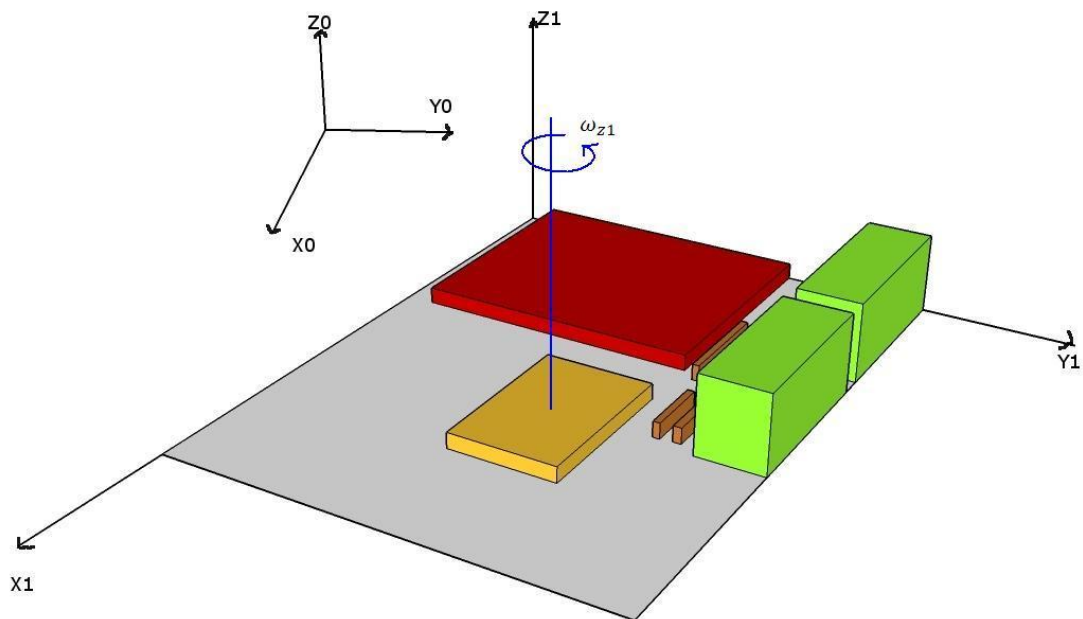
- Caricamento file
- Definizione parametri. Vengono definiti i parametri che verranno utilizzati nel resto del programma. Tra questi gli angoli e le matrici di rotazione relative alle prove fatte (vd sopra).
- Calcolo della tensione media. Viene effettuata la media di ognuno dei tre segnali relativi ai tre assi dell'accelerometro.
- Calcolo tensione calcolata. (vd sopra)
- Calcolo accelerazione misurata. (vd sopra)
- Calcolo accelerazione calcolata. Semplicemente si costruisce una matrice  $4 \times n$  con  $n$  numero di misure effettuate, dove i componenti della matrice in colonna sono  $\begin{pmatrix} g \\ 1 \end{pmatrix}_n$ .
- Stima degli errori. Si calcola la differenza fra tensioni/accelerazioni calcolate e misurate, per stimarne l'errore medio.

I risultati vengono semplicemente stampati a video, e sono riportati in tabella:

Prova	A s s e	$V_m$	$V_c$	<i>Errore tensioni</i>	$g_m$	$g_c$	<i>Errore accelerazioni</i>
1	x	1.9082	1.9060	0.0022	0.0321	0	0.0321
	y	1.0962	1.0870	0.0093	-9.6825	-9.8100	<b>0.1275</b>
	z	1.9337	1.9297	0.0041	-0.0544	0	0.0544
2	x	1.8954	1.8932	0.0022	0.0319	0	0.0319
	y	2.5385	2.5298	0.0086	9.9290	-9,810	<b>0.1190</b>
	z	1.9416	1.9375	0.0041	-0.0548	0	0.0548
3	x	1.9035	1,9074	0.0040	-0.0332	0	0.0332
	y	2.3730	2,3372	0.0358	7.4484	6,9367	0.5117
	z	1.4705	1,4203	0.0502	6.2652	6,9367	<b>0.6715</b>
4	x	1,8794	1,9165	0,0370	-0.5077	0	0.5077
	y	1,3135	1,3170	0,0034	-6.9774	-6,9367	0.0407
	z	1,4108	1,4147	0,0039	6.9941	6,9367	0.0574
5	x	2,3602	2,4203	0,0600	6.1013	6,9367	<b>0.8354</b>
	y	1,8275	1,8358	0,0083	-0.1211	0	0.1211
	z	1,3687	1,4126	0,0438	7.5339	6,9367	0.5972
6	x	1,4533	1,4036	0,0496	-6.2728	-6,9367	0.6639
	y	1,8559	1,8184	0,0375	0.4722	0	0.4722
	z	1,3684	1,4224	0,0539	7.6587	6,9367	<b>0.7220</b>
7	x	2,6273	2,6184	0,0089	9.9305	9,810	<b>0.1205</b>
	y	1,8283	1,8207	0,0075	0.0981	0	0.0981
	z	1,9210	1,9266	0,0056	0.0748	0	0.0748
8	x	1,1903	1,1807	0,0096	-9.6797	-9,810	<b>0.1303</b>
	y	1,8036	1,7960	0,0076	0.0986	0	0.0986
	z	1,9349	1,9405	0,0055	0.0745	0	0.0745
9	x	1,9057	1.8821	0.0237	0.3284	0	0.3284
	y	1,8183	1.7819	0.0365	0.4908	0	<b>0.4908</b>
	z	2,6646	2.6634	0.0013	-9.8278	-9,810	0.0178

Possiamo notare come gli errori sulle accelerazioni siano sempre inferiori a  $0,9 \frac{m}{s^2}$ , e che l'errore maggiore (per ogni misura evidenziato in grassetto) è distribuito piuttosto uniformemente sui tre assi. Questo indica che più che a problemi di taratura esso può essere accreditato a errori, comunque relativamente piccoli, di inclinazione o accelerazione su sensore al momento della misura.

## Taratura del giroscopio



La taratura del giroscopio si effettua in condizioni dinamiche, diverse da quelle dell'accelerometro, dato il principio fisico su cui si basa. Il corpo viene ruotato attorno all'asse sensibile del giroscopio ( $z_1$ ) di un angolo noto, imponendo in questo modo una velocità angolare al sensore, che viene misurata. Ovviamente per conoscere la velocità angolare, noto l'angolo, è ovviamente importante conoscere il lasso temporale in cui si effettua la misura.

I parametri che si trovano tramite la taratura sono la sensibilità  $S$  e il valore di offset  $O$  da utilizzare nell'equazione vista in precedenza:

$$V = S \cdot (\omega_{z1}) + O$$

Come abbiamo visto prima, il segnale in uscita del giroscopio è legato direttamente alla velocità angolare che subisce ( $\omega_{z1}$ ). Conoscendo gli istanti di inizio e fine rotazione ( $t_1$  e  $t_2$ ) è possibile integrare il segnale campionato, ottenendo così l'angolo di rotazione. In formule

$$\theta_z = \int_{t_1}^{t_2} \frac{(V - O)}{S} dt \Rightarrow S = \frac{1}{\theta_z} \int_{t_1}^{t_2} (V - O) dt$$

Ovvero conoscendo  $V$  e  $\theta_z$  (l'angolo di rotazione, noto) possiamo calcolare la sensibilità. Per calcolare l'offset si può invece usare uno stratagemma, tenere fermo il giroscopio all'inizio della misura: mediando il valore della  $V$  rilevata prima dell'inizio della rotazione otterremo proprio il valore di offset.

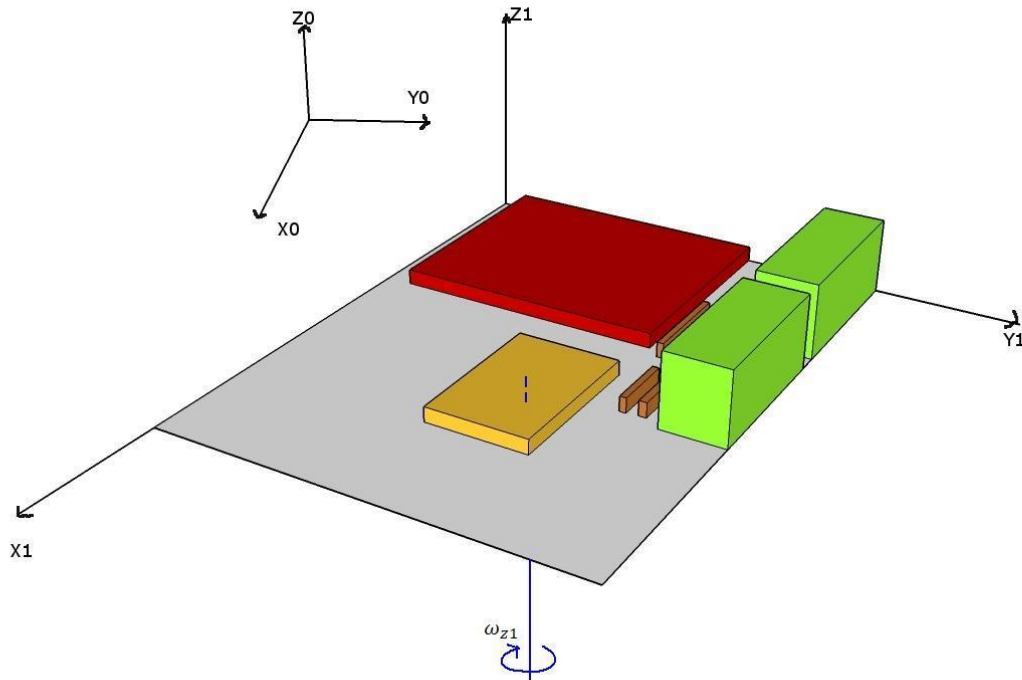
Non essendo il segnale in uscita dal giroscopio continuo, ma campionato, è necessario effettuare il calcolo di  $\omega_z$  punto per punto. Avremo così una serie di misure che, definito per comodità  $W$  l'integrale:

$$W = \int_{t_1}^{t_2} (V - O) dt$$

Porterà a una serie di misure che ci portano un singolo valore di sensibilità attraverso:

$$S = \text{pinv} \begin{pmatrix} \theta_1 \\ \dots \\ \theta_n \end{pmatrix} \cdot (W_1 \quad \dots \quad W_n)_{(n \times 1)}^T$$

Da notare che in fase realizzativa il giroscopio è stato montato con l'asse z rivolto verso il basso rispetto al riferimento  $xyz_0$  e le rotazioni risulteranno quindi invertite. In particolare, ci aspettiamo di riscontrare una sensibilità negativa rispetto a quella su datasheet (semplicemente perché le rotazioni in senso orario su  $xyz_1$  risulteranno in senso antiorario sul giroscopio), mentre l'offset dovrebbe restare invariato.



## Realizzazione della Taratura

Per effettuare la taratura effettuiamo 5 rotazioni, attendendo alcuni secondi durante il campionamento prima di effettuare la rotazione, per poter calcolare l'offset. Per positiva si intende una rotazione in senso orario rispetto al riferimento  $xyz_0$ .

1.  $+90^\circ$
2.  $-90^\circ$
3.  $-180^\circ$
4.  $-45^\circ$
5.  $+45^\circ$

Una volta ottenuti i dati, tramite interfaccia LabView, possiamo effettuare il post processing tramite Matlab.

Il programma in appendice effettua proprio questo lavoro. Lo schema di funzionamento:

- Import dei dati e caricamento dei parametri

- **Filtraggio.** Viene applicato un filtro passabasso di ordine 2 con frequenza di taglio di 10Hz. L'implementazione in Matlab viene fatta tramite la funzione *butter*. Il primo parametro che viene passato a questa funzione indica l'ordine del filtro, mentre il secondo il rapporto tra frequenza di taglio voluta e frequenza di Nyquist (che per il nostro campione, avendo frequenza di campionamento di 1000 Hz equivale a 500 Hz). Siccome un filtraggio normalmente implica anche una traslazione temporale della fase del segnale si è scelto di utilizzare la funzione *filtfilt* che applica una successiva traslazione "all'indietro", riportando la fase del segnale al valore iniziale. In questo modo si ottiene un filtro a fase zero. Il filtro serve ad eliminare il rumore ad alta frequenza, causato ad esempio da un'alimentazione non corretta.
- **Calcolo Offset.** Per calcolare il valore di offset come visto utilizziamo la prima parte del file dati, quando cioè non è ancora iniziata la rotazione. L'inizio di questa viene preso da quando la variazione di tensione supera il 18% del valore massimo. Per calcolare l'offset si media il segnale fino a quel punto.
- **Calcolo fine curva.** Con un procedimento simile, ma partendo da fine file e andando a ritroso, calcoliamo il punto di fine curva.
- **Calcolo sensibilità.** Conoscendo gli offset possiamo calcolare V-O e integrare il segnale per ottenere

$W$ . Per ottenere la sensibilità applichiamo  $S = pinv \begin{pmatrix} \theta_i \\ \dots \\ \theta_n \end{pmatrix} \cdot (W_1 \dots W_n)_{(n \times 1)}^T$  dove ora sia  $\theta$  che  $W$  sono noti.

## I risultati

- Sensibilità  $S = -0.0026$
- Offset  $O = 2.2045V$

Come ci attendevamo la sensibilità risulta negativa. I risultati non sono in linea con quelli indicati dal datasheet, che prevede come offset 1.65 V e come sensibilità  $3.3 \frac{V \cdot s}{^\circ}$  (corrispondente alla variazione di 1/3 di grado sessagesimale per ogni variazione di 1mV). Valori così differenti potrebbero esser stati causati da accoppiamenti sul circuito di tipo resistivo ad esempio, portando la tensione di alimentazione del giroscopio a un valore maggiore di quello atteso. È possibile che questo stesso problema abbia modificato leggermente anche il valore della sensibilità ma non abbiamo modo di accertarcene. Sarebbe opportuno verificare se i contatti, in particolare massa e alimentazione, del giroscopio sono stati eseguiti correttamente in fase di collegamento. A causa di questi valori di Sensibilità e offset ci aspettiamo possibili problemi di riscontro dati nei prossimi test che verranno svolti nel proseguo del lavoro.

## Verifica del Giroscopio

Per verificare il giroscopio verranno effettuate due prove. La prima è equivalente a quella fatta per l'accelerometro: effettueremo misure diverse da quelle usate in taratura, e verificheremo il grado di vicinanza tra l'angolo noto e quello calcolato; questo ci permette di verificare se i parametri di offset e Sensibilità sono corretti, test particolarmente importante per noi, visti i valori ottenuti precedentemente. La seconda verifica sarà volta a riscontrare l'entità dell'effetto di deriva. Questa verrà effettuata ruotando il sensore in un intervallo di angoli noti, per un tempo sufficientemente lungo.

### REALIZZAZIONE – Prima prova

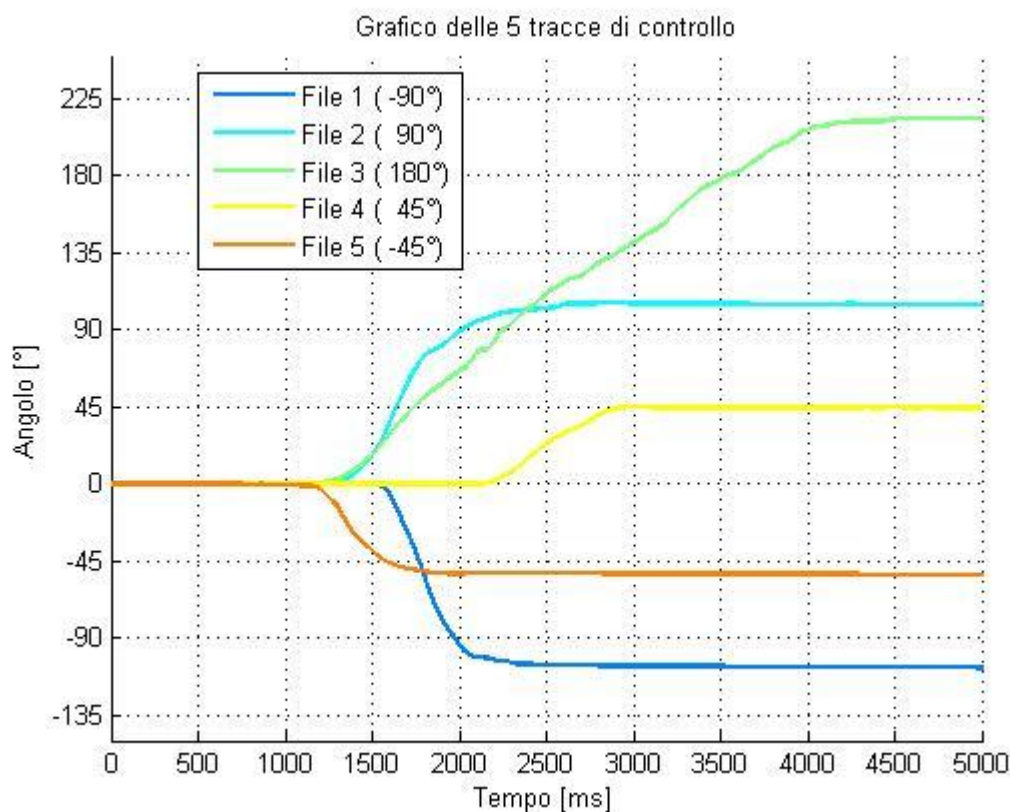
Per la prima prova le 5 rotazioni imposte sono le seguenti:

- 1)  $-90^\circ$
- 2)  $90^\circ$

- 3) 180°
- 4) 45°
- 5) -45°

Anche in questo caso, una volta acquisiti i dati tramite LabView, si utilizza un software Matlab, il cui listato è presente in appendice. Il programma è suddiviso in questo modo:

- Import dati e parametri. All'interno di questa sezione viene effettuato anche il filtraggio dei dati come visto in precedenza.
- Angolo calcolato. In questa parte del programma viene calcolato l'angolo a partire dai dati forniti dal giroscopio. Si effettua un'integrazione tramite la funzione *trapz* del file già filtrato (V-O) e si divide per la sensibilità ( $\theta_z = \int_{t_1}^{t_2} \frac{(V-O)}{S} dt$ ).
- Errore commesso. Viene stimato l'errore tra l'angolo noto e quello calcolato. L'errore viene riportato in percentuale.
- Stampa dei dati. Viene creato un grafico dove sono presenti, con colori diversi e legenda, tutte le tracce riguardanti gli angoli (in gradi), mentre gli errori sono stampati a video. I risultati sono riportati in figura e gli errori in tabella.



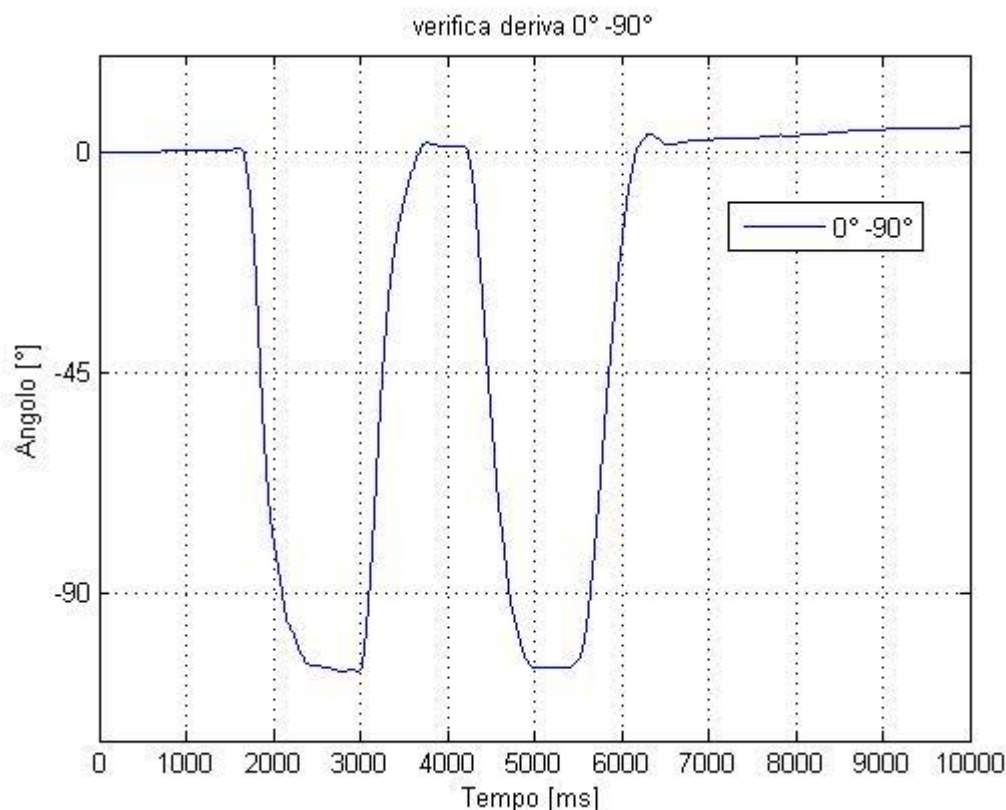
Rotazione	Errore
-90°	19.27%
90°	17.56%
180°	18.58%
45°	0.17%
-45°	18.31%

Gli errori commessi sono relativamente ampi. Questo può essere imputato sia ai parametri di taratura, come avevamo d'altronde preventivato, sia alla difficoltà di effettuare a mano queste rotazioni. È invece positivo notare che la presenza di deriva è quasi inesistente; questo ci rassicura circa le misure effettuate col giroscopio per tempi ridotti (max 5 secondi).

## REALIZZAZIONE – Seconda prova

Ora andremo ad analizzare il fenomeno della deriva per intervalli di tempo più lunghi. Verrà eseguita una rotazione tra  $0^\circ$  e  $+90^\circ$ , per un tempo di 10 secondi. Come prima utilizzeremo un'altra applicazione sviluppata in Matlab. Ecco lo schema di funzionamento.

- Import di dati e parametri. All'interno di questa sezione viene effettuato anche il filtraggio dei dati come visto in precedenza.
- Soluzione. Viene calcolato l'angolo  $\theta_z$  tramite integrazione del segnale
- Calcolo errore. Calcoliamo l'errore massimo percentuale commesso. L'errore è calcolato considerando che l'angolo dovrebbe essere compreso tra  $0^\circ$  e  $90^\circ$ .
- Stampa dei dati. Vengono presentati i dati su grafico. I risultati sono riportati in figura.



L'errore massimo stimato è del 17.52%, in linea comunque con quelli visti in precedenza. Un aspetto che ci è sembrato interessante è l'ultima parte del grafico ovvero per tempi superiori ai 7 secondi (7000 ms sul grafico). Nonostante il giroscopio sia stato tenuto pressoché fermo, il valore dell'angolo stimato cresce, con un andamento che è tipico dei fenomeni di deriva. Si è deciso di stimare anche questo errore, che risulta essere, nel punto peggiore, di  $5.49^\circ$ , un valore non trascurabile.

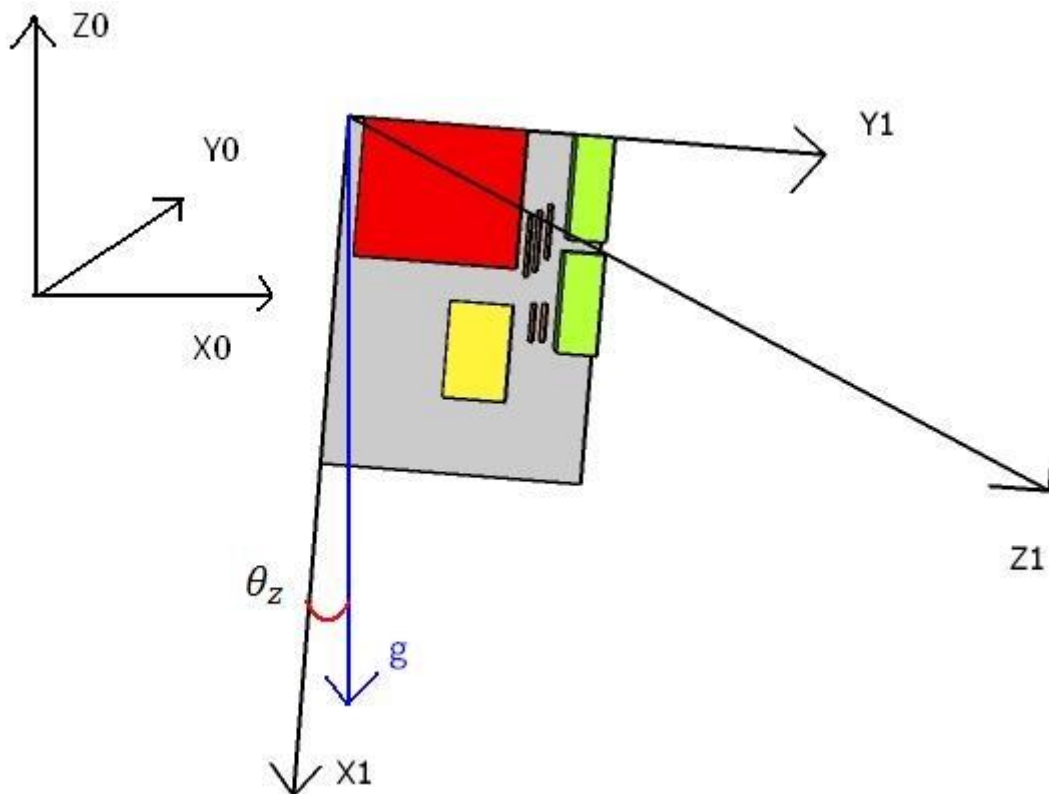
Mentre l'errore massimo può ancora essere attribuito a problemi di taratura e setup della misura, l'errore più riconoscibile come colpa della deriva è quello relativo all'ultima parte della misura. Da notare che il fenomeno sembra accentuarsi a giroscopio fermo, il che ci consiglia particolare attenzione quando si volesse utilizzare il sensore in condizioni statiche o di bassa rotazione.

## CALCOLO ANGOLI DI ROTAZIONE

Come visto nell'ultima verifica, soprattutto per tempi di acquisizione lunghi, l'angolo calcolato tramite giroscopio ha bisogno di una correzione soprattutto per la deriva. Esistono molte tecniche a livello industriale, e tra queste molte richiedono algoritmi costosi, che alzano notevolmente il prezzo del sensore. Anche noi abbiamo deciso di implementarne uno molto semplice basato sul segnale dell'accelerometro.



L'algoritmo che utilizzeremo è quello semplificato visto durante il corso. Ricordiamo brevemente i passaggi utilizzati, in modo da rendere più comprensibile la lettura del file Matlab che esamineremo tra poco.



Ipotizziamo che l'IMU si trovi con l'asse z parallelo al suolo e l'asse x (ad esempio) perpendicolare ad esso e rivolto verso il basso, come in figura. In questo caso l'accelerometro avrà due assi sensibili su g: x e y. Tramite le tensioni misurate sui due assi possiamo calcolare l'asse dell'accelerometro  $n_{acc}$ , sfruttando l'offset e la sensibilità calcolate in precedenza. Ovviamente il versore iniziale dipenderà strettamente dalla posizione iniziale degli assi, e sarà infatti il primo controllo che verrà fatto dalla nostra applicazione (asse x verso l'alto o verso il basso, stessa cosa per l'asse y e verso di rotazione iniziale). Considerando l'asse x che punta verso il basso e verso di rotazione antiorario il versore avrà questa forma:

$$\hat{n}_{acc} = \begin{pmatrix} -\cos(\theta_{acc}) \\ -\sin(\theta_{acc}) \\ 0 \end{pmatrix}$$

Con  $\theta_{acc}$  l'angolo calcolato tramite l'accelerometro, che a questo punto possiamo calcolare tramite:

$$\theta_{acc} = \arcsin\left(\frac{-a_x}{g}\right)$$

La stessa forma avrà ovviamente il versore del giroscopio:

$$\hat{n}_{gir} = \begin{pmatrix} -\cos(\theta_{gir}) \\ -\sin(\theta_{gir}) \\ 0 \end{pmatrix}$$

E l'angolo verrà calcolato come visto in precedenza:

$$\theta_{gir} = \int_{t_1}^{t_2} \frac{(V - O)}{S} dt$$

Visto che l'accelerometro è esente da deriva (ma non da rumore) non essendo necessaria l'integrazione del segnale si potrebbe pensare di utilizzare il solo accelerometro per stimare l'angolo di rotazione. Questa considerazione è sicuramente giusta per rotazioni lente, ma non per rotazioni veloci. Ricordiamo infatti che la condizione di utilizzo dell'accelerometro come inclinometro è limitata al regime di quasi-staticità della misura, in quanto  $a$  deve essere molto minore di  $g$ . Questo non è sicuramente vero per rotazioni veloci, che

richiedono l'imposizione di una certa accelerazione non trascurabile alla IMU. Si rende necessario quindi affiancargli il giroscopio, per giunta già presente sulla IMU. Questo strumento però si porta dietro tutti i problemi di deriva già noti, che possiamo correggere con l'accelerometro con il seguente algoritmo. Si calcola l'angolo  $\delta$  compreso tra i due vettori individuati:

$$\delta = \arccos \frac{n_{acc} \cdot \hat{n}_{gir}}{|n_{acc}| |\hat{n}_{gir}|}$$

Dove  $|n_{acc}|$  e  $|\hat{n}_{gir}|$  sono la norma del vettore e  $\cdot$  è il prodotto scalare tra vettori.

Si moltiplica questo angolo per un fattore correttivo  $\lambda$  ottenendo la correzione per la velocità angolare.

$$\omega_{cor}^{(N)} = \lambda \cdot \delta^{(N)}$$

$\lambda$  può essere considerato il "grado di fiducia" che assegniamo alla misura del giroscopio: a seconda che il suo valore sia più o meno alto, daremo maggior importanza ai dati misurati dall'accelerometro o a quelli del giroscopio. Nota la correzione è possibile calcolare la nuova velocità angolare:

$$\omega_y^{(N+1)} = \omega_{gir}^{(N+1)} - \omega_{cor}^{(N)} = \lambda \cdot \delta^{(N)}$$

Da notare che per correggere il passo presente, si utilizza la  $\omega$  di correzione al passo precedente.

Si ripete l'iterazione e infine si integra il segnale.

Un aspetto da sottolineare: la posizione del sistema di riferimento  $xyz_1$  non è trascurabile. Se infatti la IMU fosse posta con l'asse  $z_1$  rivolto perpendicolarmente al suolo, questo algoritmo semplificato non sarebbe applicabile. Come abbiamo visto nella sezione relativa infatti, l'accelerometro non può funzionare da inclinometro per rotazioni attorno all'asse  $z_0$ : essendo la forza dovuta al vettore  $g$  perpendicolare ai due assi sensibili  $x_1$  e  $y_1$  e parallela a  $z_1$ , una rotazione attorno a quest'ultimo non porta cambiamenti nella misura da parte del sensore, e quindi non dà nessuna informazione a riguardo.

## REALIZZAZIONE

A causa di problemi legati a un non perfetto contatto è stato impossibile utilizzare le misure eseguite con la IMU presentata finora. Si è pertanto scelto di utilizzare un'altra IMU pressoché identica al fine di mostrare comunque il funzionamento dell'algoritmo e i risultati espressi. La nuova IMU varia i parametri di taratura del giroscopio come segue:

- *Sensibilità* = -0.0033
- *Offset* = 1.66

I parametri di taratura dell'accelerometro sono pressoché uguali e non vengono riportati.

Verranno eseguite 6 misure:

- 1) Rotazione lenta di 90° per 30 secondi
- 2) Rotazione media di 90° per 30 secondi
- 3) Rotazione media di 90° per 30 secondi
- 4) Rotazione lenta di 180° (-90° +90°) per 60 secondi
- 5) Rotazione media di 180° per 60 secondi
- 6) Rotazione veloce di 180° per 60 secondi

Facendo delle considerazioni preliminari, considerando quanto visto in precedenza, è facile aspettarci una maggiore deriva per i dati raccolti in un tempo maggiore, grado di attendibilità migliore per l'accelerometro rispetto al giroscopio per rotazioni lente ( $\lambda$  alto) e viceversa per rotazioni veloci ( $\lambda$  più basso).

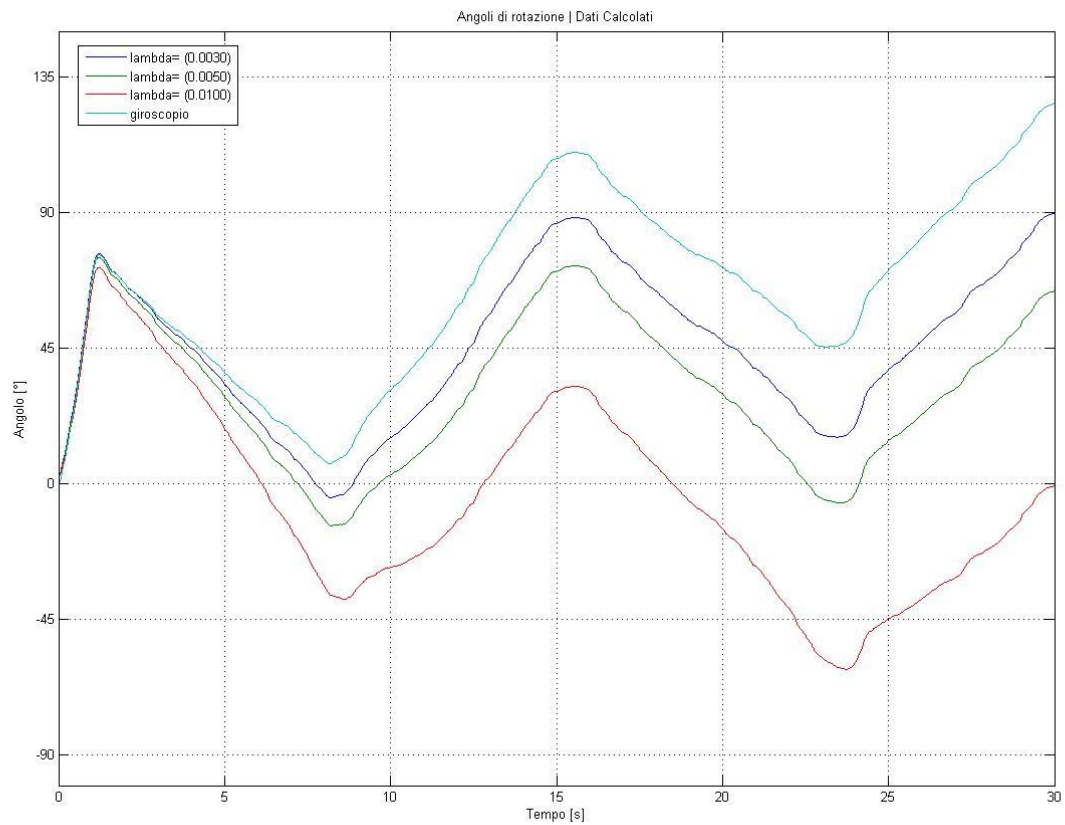
L'algoritmo utilizzato è stato pensato per gestire file di medie dimensioni, come è facile aspettarci per acquisizioni di più di 30 secondi, ed è stato suddiviso in funzioni per renderne più accessibile la lettura e una eventuale successiva modifica.

- Il *main program* ha funzione principalmente di import dati e caricamento file. Viene caricato solo un singolo file e non tutti insieme perché la gestione di tutte le sei misure in fila richiederebbe un tempo di computazione molto elevato. Per lo stesso motivo è stato implementato un sistema per ridurre in post processing la frequenza del segnale, che permette di processare un numero molto minore di dati, mantenendo una buona approssimazione. Questo artificio non è stato poi in effetti necessario visto che il calcolatore utilizzato, rispetto alla frequenza di campionamento, ha permesso un tempo di computazione ancora ragionevole, pur mantenendo la frequenza di campionamento originale. Una volta caricati i dati il programma effettua il filtraggio degli stessi. A questo punto chiama la funzione *calcola\_angolo* dalla quale ottiene l'angolo finale già corretto. Infine richiama la funzione *stampa\_grafici* per mostrare i risultati ottenuti graficamente.
- La funzione *calcola\_angolo* prende come input oltre ai dati già filtrati, anche i dati di taratura di accelerometro e giroscopio, e il vettore  $\lambda$ , contenente tre valori di  $\lambda$  da testare. In questo modo esamineremo anche quale possa essere il  $\lambda$  migliore per quella determinata misura. Come output ritorna l'angolo di rotazione già corretto. La funzione per prima cosa calcola il versore dell'accelerometro dividendo le accelerazioni misurate dallo strumento (tramite i parametri di taratura) per il modulo (la sensibilità), e quindi l'angolo calcolato dall'accelerometro punto per punto tramite la formula già vista:  $\theta_{acc} = \arcsin\left(\frac{-a_x}{g}\right)$ . La funzione calcola l'angolo misurato dal giroscopio tramite i parametri di taratura esattamente come visto in precedenza. A questo punto richiama la funzione *calcola\_versore* per calcolare la forma del versore del giroscopio punto per punto. Noti a questo punto entrambi i versori calcola la  $\delta$  di correzione e la moltiplica volta per volta per i tre valori di  $\lambda$ . Per ognuno di questi valori viene corretta la velocità angolare del giroscopio e calcolato l'angolo di rotazione corretto, che viene rilanciato al programma main.
- La funzione *calcolo\_versore* si occupa di calcolare il versore del sistema di riferimento  $xyz_1$ . Come input richiede i dati filtrati, l'angolo calcolato dal giroscopio e i parametri di taratura dell'accelerometro. Come output ritorna il versore punto x punto. Per ogni punto la funzione valuta la posizione degli assi x e y: ad esempio se  $x > y$  e di  $O_x$  indica che x è diretto verso il basso perché sull'asse sensibile è presente un'accelerazione pari a +g; inoltre calcola il verso di rotazione iniziale, considerata tra il primo e il 50° punto.
- La funzione *stampa\_grafici* si occupa di stampare a video i risultati, formattandoli correttamente. In input prende una matrice M contenente i dati da stampare, l'angolo calcolato dal solo giroscopio, il titolo, il numero della figura e i valori di  $\lambda$  utilizzati.

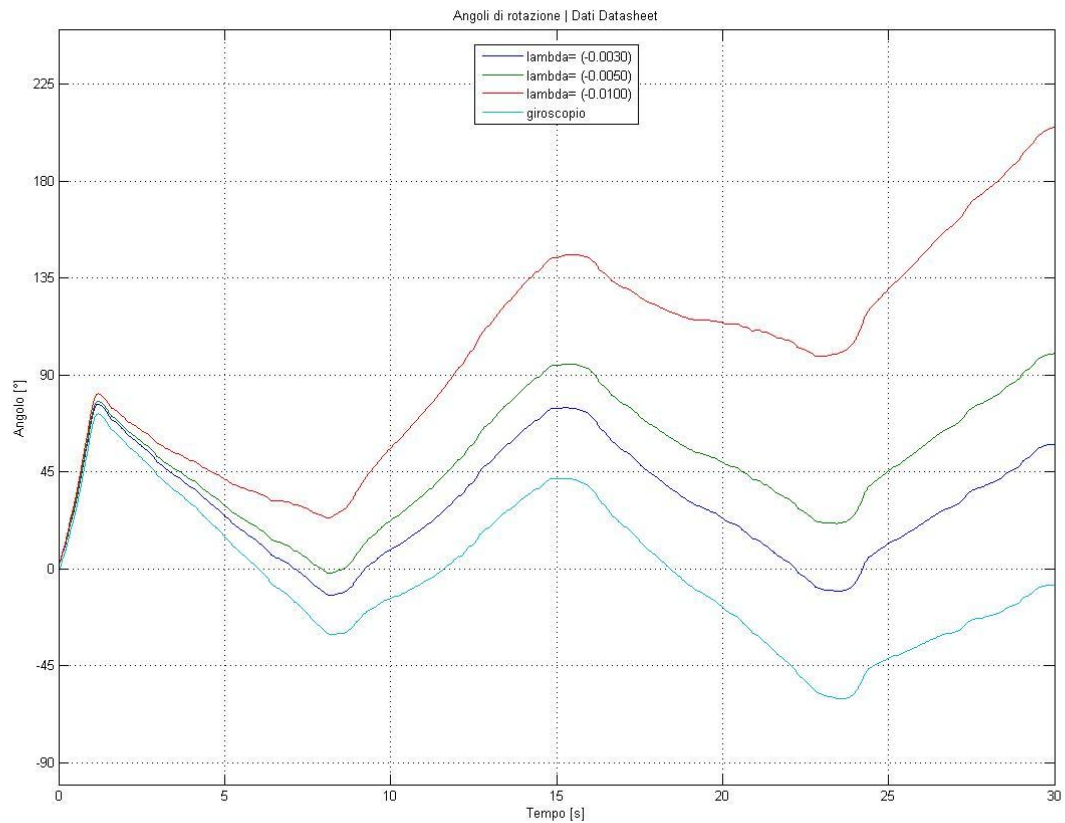
Di seguito mostreremo e discuteremo i risultati ottenuti. Riporteremo i dati relativi sia ai parametri di taratura calcolati, che a quelli presi da datasheet per confronto.

### **Rotazione tra 0 e -90° lenta:**

In figura vediamo sia la traccia del giroscopio che le tracce corrette con  $\lambda$  diversi. Si sono scelti come valori di  $\lambda$  0.003, 0.005 e 0.01. Questi sono stati scelti testando preventivamente molti valori, cercando quali fossero i  $\lambda$  che meglio correggevano la traccia del giroscopio. Questi tre sono risultati i migliori sulle varie misure. Tra questi di volta in volta verrà scelto il migliore. Inoltre una volta eseguito il test con i parametri di taratura calcolati precedentemente, verranno mostrati per confronto i dati ottenuti tramite parametri da datasheet.

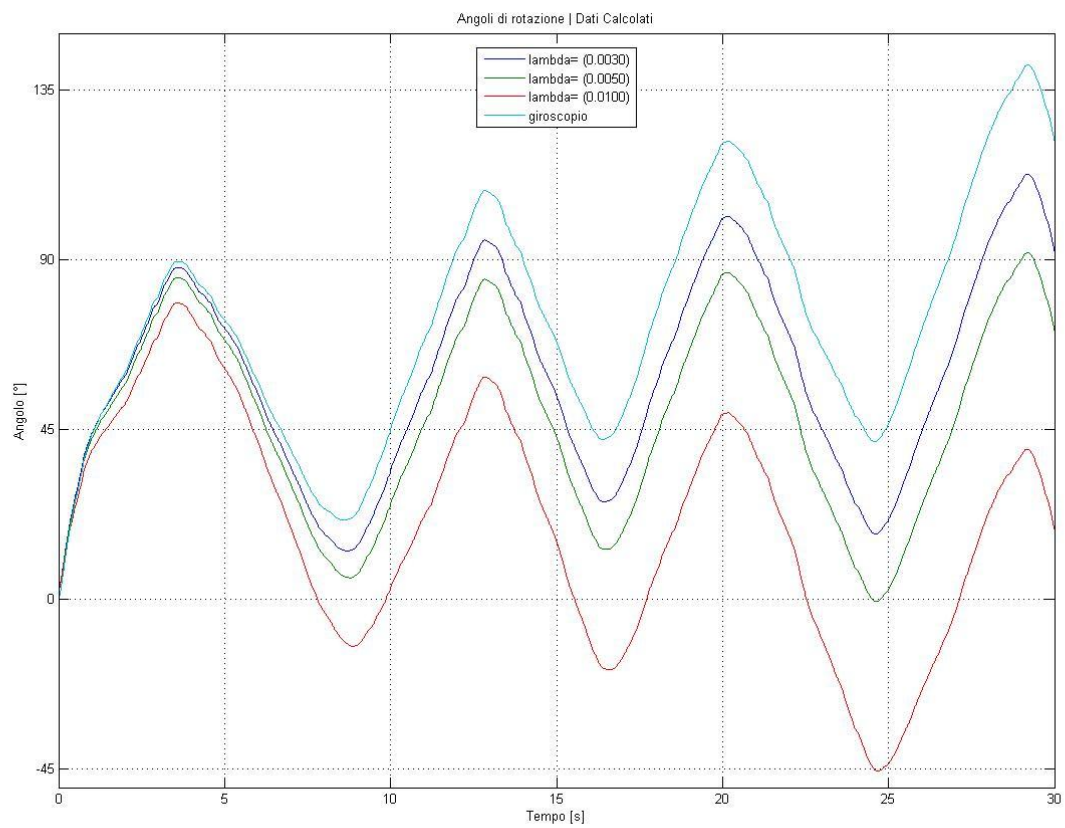


In figura è molto chiaro come l'angolo calcolato tramite giroscopio soffra pesantemente il fenomeno di deriva. Effettuando la correzione notiamo come l'angolo migliori notevolmente. Il valore di  $\lambda$  ideale, in buona approssimazione, risulta 0.005, anche se probabilmente un  $\lambda$  migliore potrebbe essere trovato tra 0.003 e 0.005.

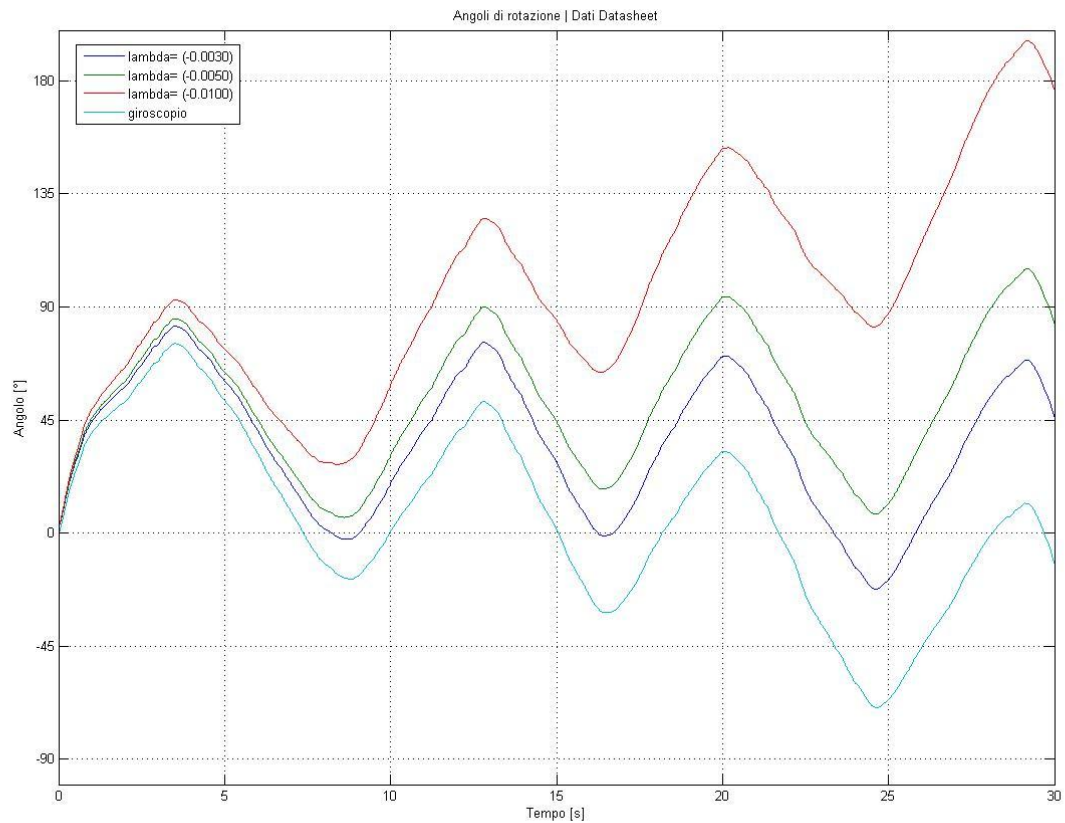


Si nota come usando i dati del datasheet i risultati sono peggiori. Questo dimostra l'utilità dell'operazione di taratura. I valori di  $\lambda$  quando si considerano i dati del datasheet sono da intendersi negativi, e in questo caso il valore di  $\lambda$  migliore risulta più vicino a 0.003.

*Rotazione tra  $0^\circ$  e  $+90^\circ$  media:*



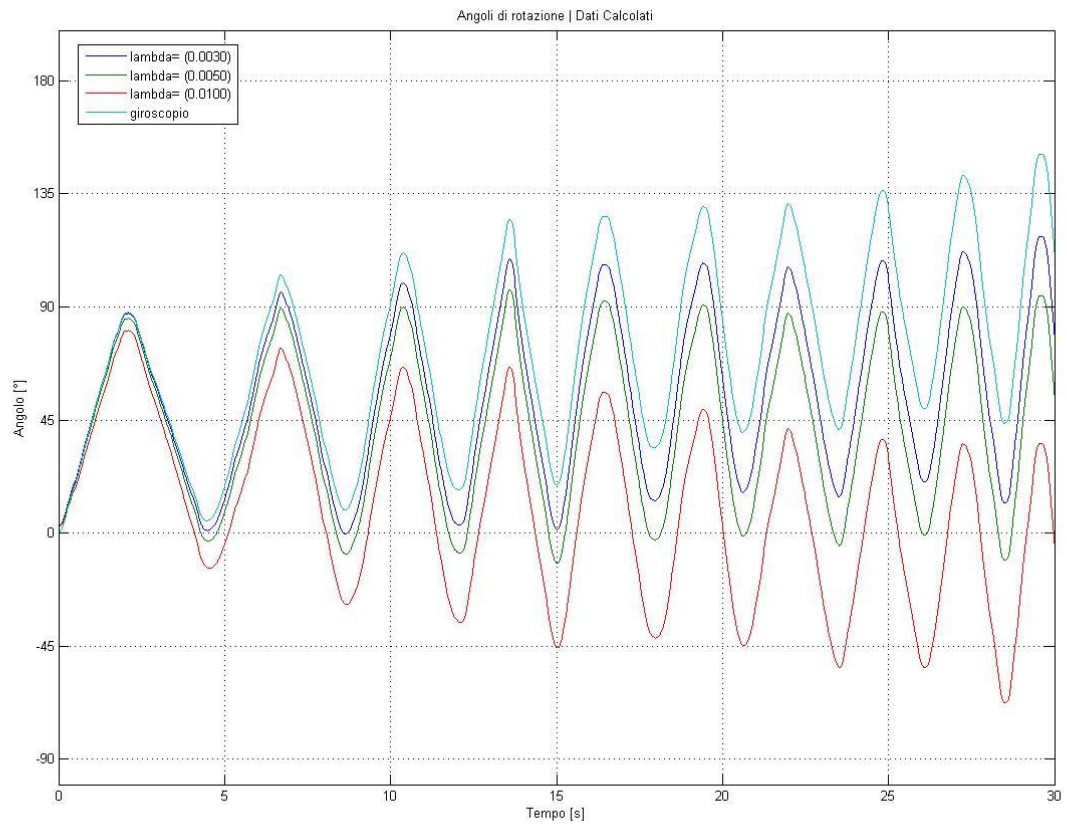
In questo caso  $\lambda$  ideale risulta 0.005.



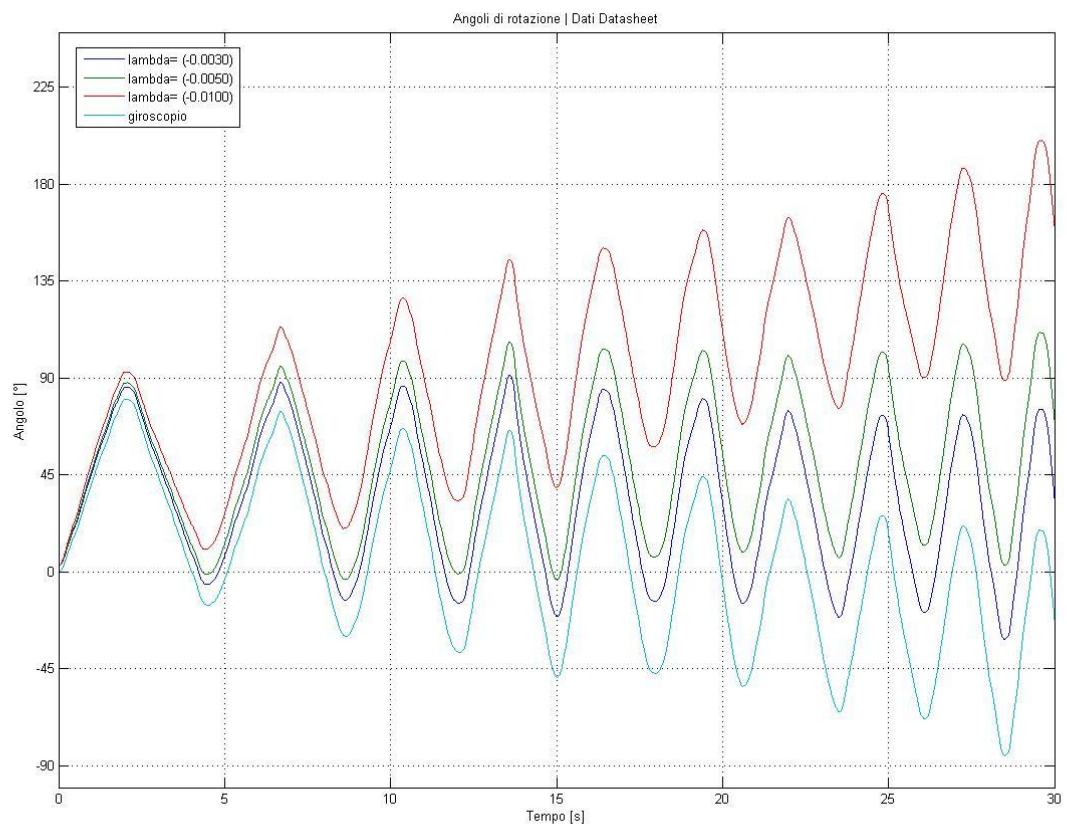
Anche in questo caso i dati presi da datasheet portano a risultati piuttosto diversi. La correzione migliore risulta ancora 0.005.

**Rotazione tra  $0^\circ$  e  $+90^\circ$  veloce:**





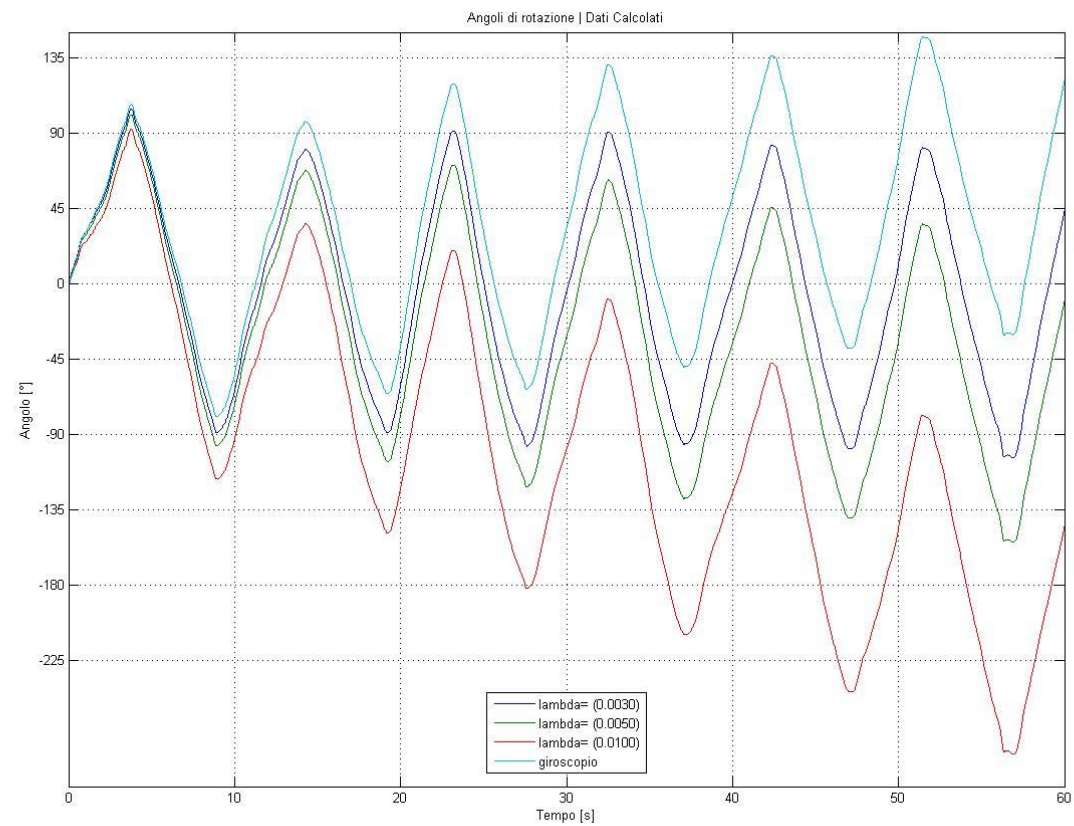
$\lambda$  ideale 0.005.



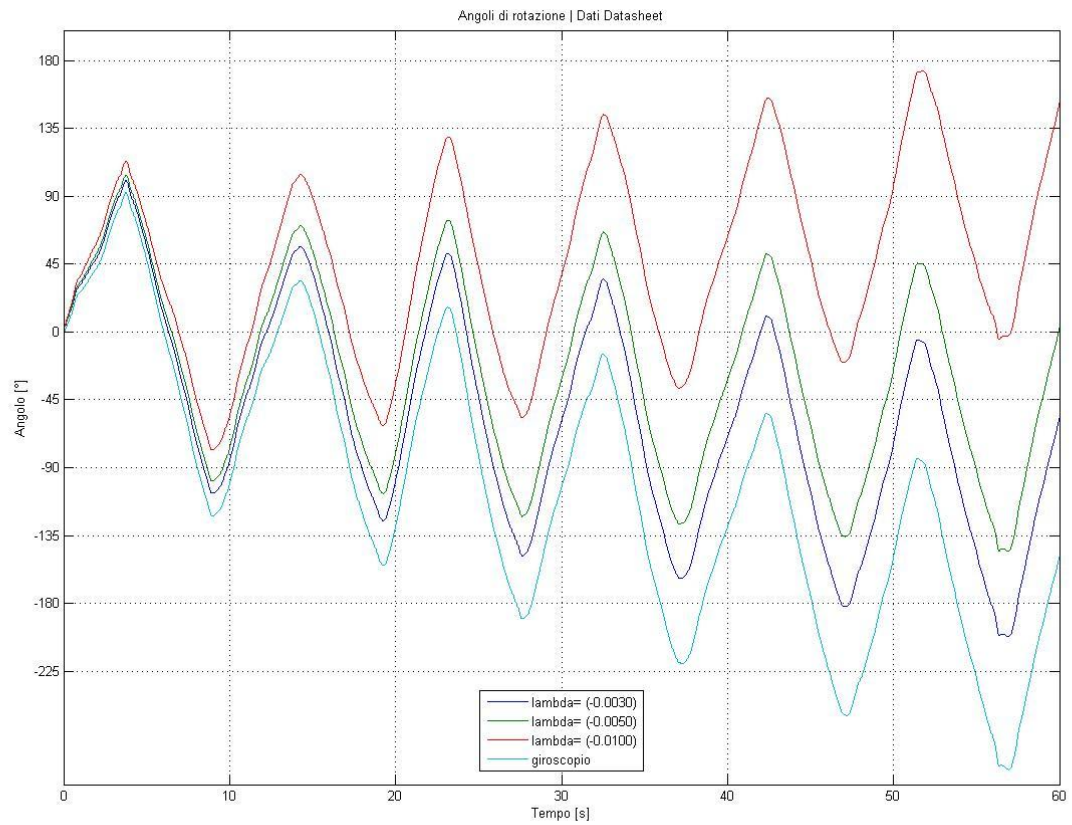


Anche qui  $\lambda$  ideale corrisponde a 0.005, soprattutto per tempi superiori ai venti secondi.

*Rotazione tra -90° e +90° bassa:*

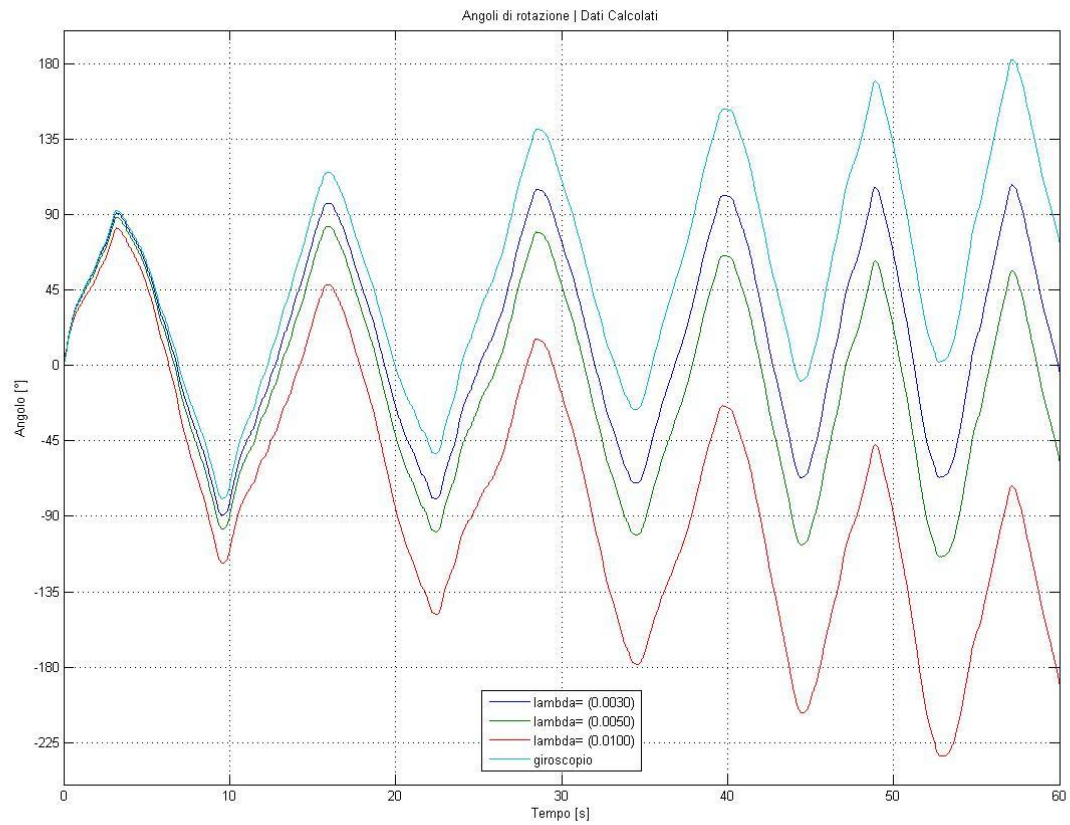


$\lambda=0.003$

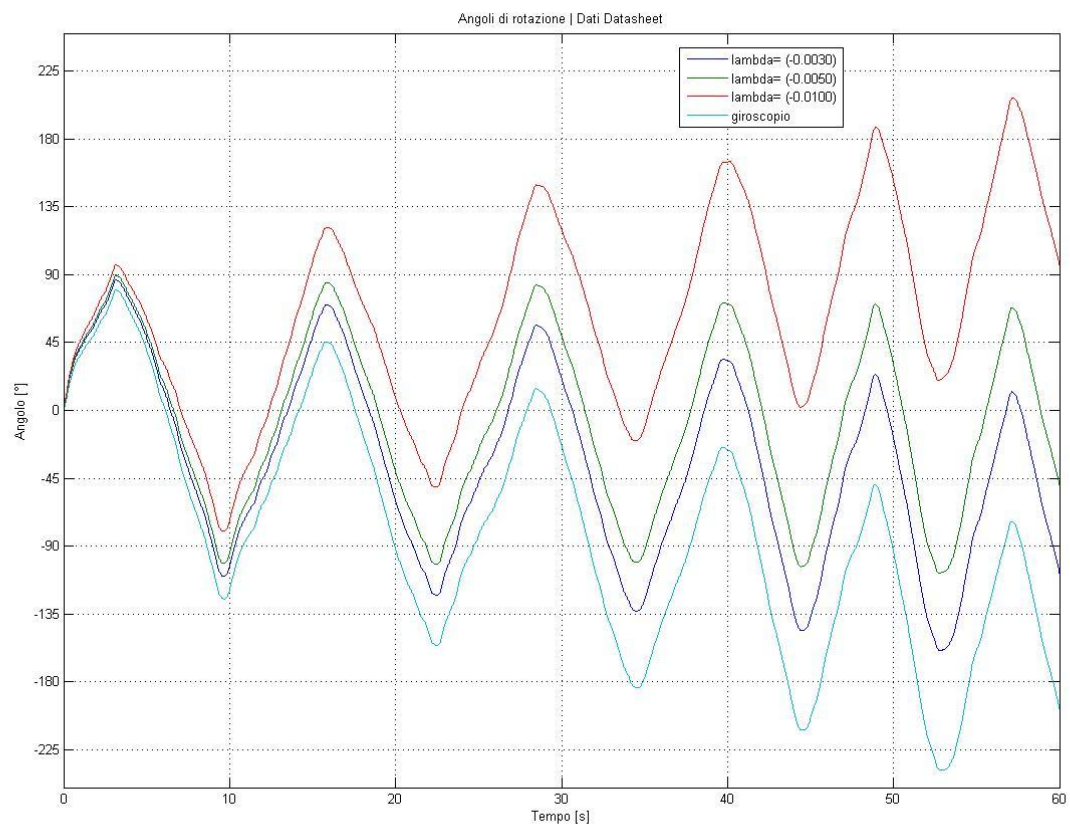


$\lambda=0.005$ .

**Rotazione tra  $-90^\circ$  e  $+90^\circ$  media:**

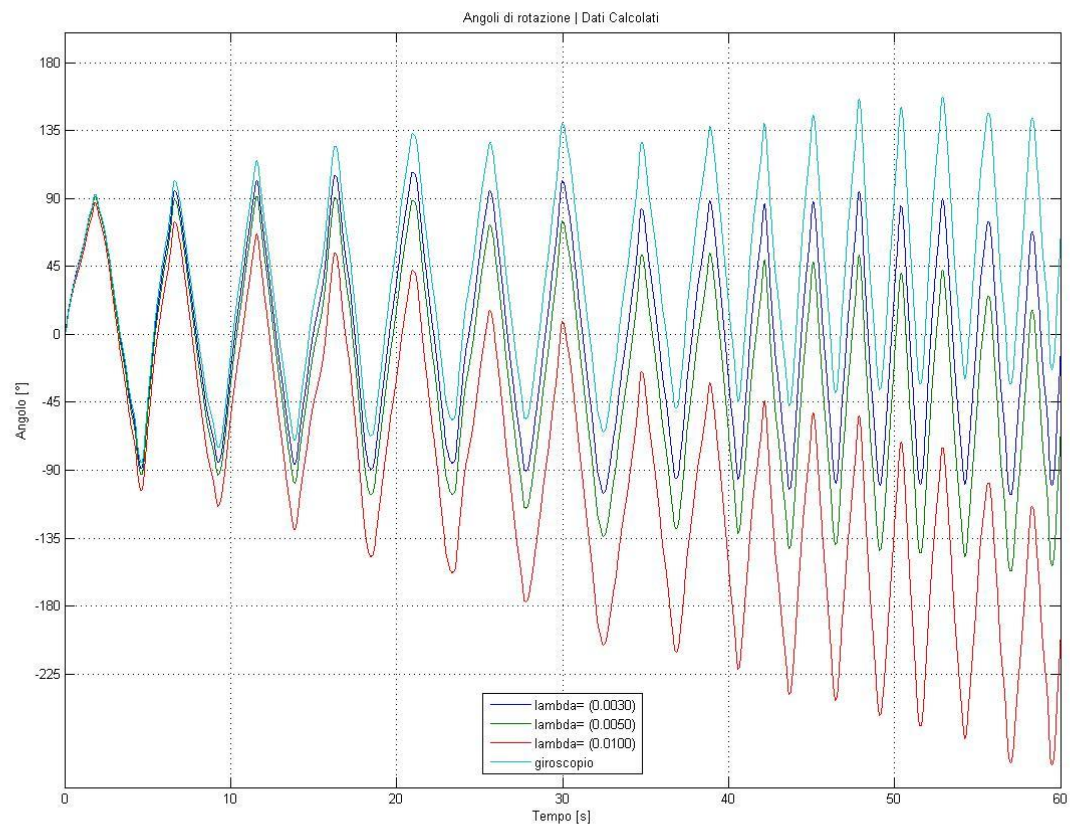


$\lambda = 0.005$ .

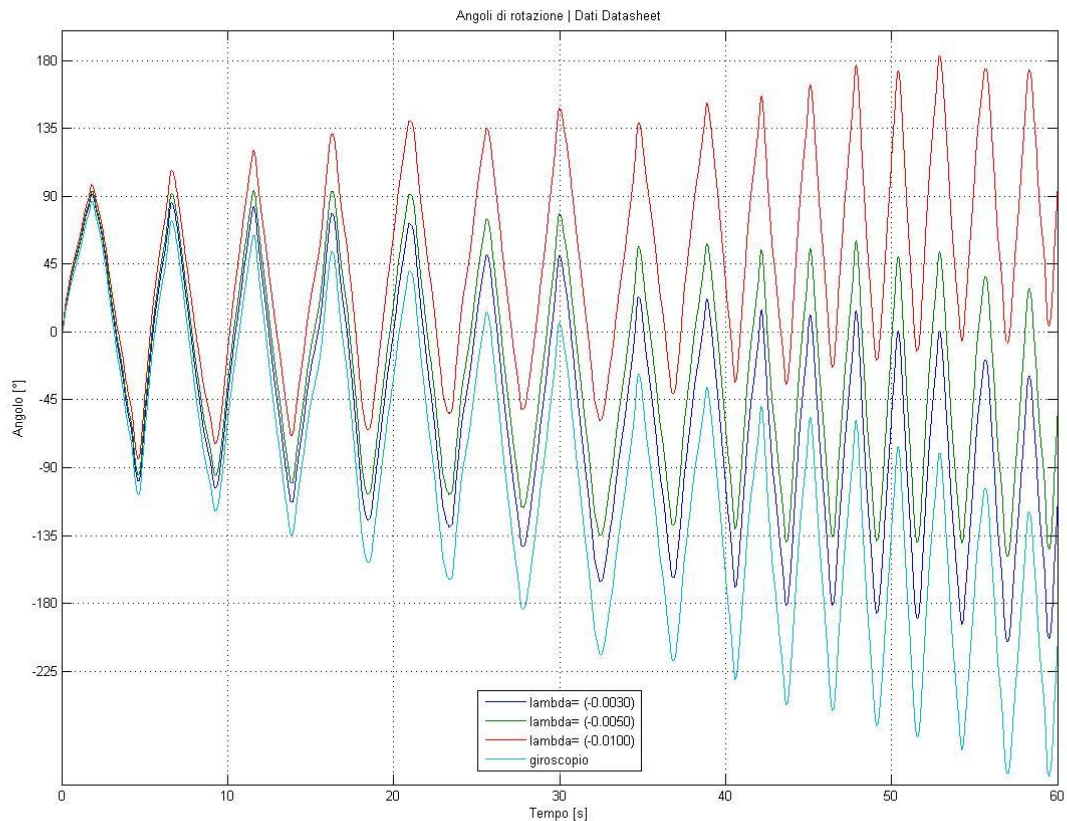


$\lambda=0.005$ .

*Rotazione tra -90° e +90° alta:*



$\lambda=0.003$ .



$\lambda=0.005$  anche se l'errore qui è piuttosto elevato, quindi sarebbe opportuno testare un valore compreso tra 0.005 e 0.01.

Questi risultati suggeriscono l'utilizzo di  $\lambda$  più elevati in caso di rotazioni lente, in modo da far aumentare l'importanza del dato proveniente dall'accelerometro, e viceversa  $\lambda$  inferiori in caso di rotazioni veloci: l'influenza dell'accelerazione inerziale può infatti inficiare la misura dell'accelerometro e suggerisce un maggior peso per quello del giroscopio. Inoltre suggeriscono di utilizzare i parametri di taratura trovati piuttosto che usare quelli presi da datasheet, che portano a errori decisamente maggiori. Questo è da imputarsi principalmente ai parametri di setup con cui è stata realizzata la IMU (collegamenti, alimentazione, ecc.) e a fattori ambientali.

## CONCLUSIONI

Abbiamo realizzato a partire dai componenti di base una IMU, con lo scopo di monitorare il percorso e la rotazione di un corpo. Una volta realizzata, è stato necessario tarare i componenti (accelerometro e giroscopio) al fine di ottenere una misura adeguata e per quanto possibile esente da errori di setup. I parametri di taratura del giroscopio sono risultati piuttosto diversi da quelli riportati nel datasheet, il che fa pensare a problemi di tipo resistivo sulla scheda.

Per assicurarci di non aver commesso errori in fase di taratura abbiamo effettuato dei test sia sui parametri di taratura trovati, che sul fenomeno di deriva che abbiamo riscontrato nei dati derivanti dal giroscopio. Infine abbiamo utilizzato un algoritmo di correzione per la deriva del giroscopio, tramite i dati dell'accelerometro, per calcolare gli angoli di rotazione imposti alla IMU. L'utilizzo combinato dei due sensori ha quindi permesso di ottenere risultati migliori, rispetto al loro utilizzo singolo.



## Listati Matlab

In questa sezione vengono presentati tutti i listati Matlab usati durante il lavoro. I listati sono corredati di alcuni commenti per renderne più semplice la lettura. Per gli schemi di funzionamento fare riferimento alle sezioni relative.

### TARATURA ACCELEROMETRO

```
clear all
%per prima cosa faccio l'import del file
file1 = fopen('C:\Documents and Settings\las.LAS-P\Desktop\Ste\nanotech new\lab
sensori\dati acquisiti(3)\0_0_0.lvm','r');
file2 = fopen('C:\Documents and Settings\las.LAS-P\Desktop\Ste\nanotech new\lab
sensori\dati acquisiti(3)\0_0_180.lvm','r');
file3 = fopen('C:\Documents and Settings\las.LAS-P\Desktop\Ste\nanotech new\lab
sensori\dati acquisiti(3)\0_-90_0.lvm','r');
file4 = fopen('C:\Documents and Settings\las.LAS-P\Desktop\Ste\nanotech new\lab
sensori\dati acquisiti(3)\0_+90_0.lvm','r');
file5 = fopen('C:\Documents and Settings\las.LAS-P\Desktop\Ste\nanotech new\lab
sensori\dati acquisiti(3)\-90_0_0.lvm','r');
file6 = fopen('C:\Documents and Settings\las.LAS-P\Desktop\Ste\nanotech new\lab
sensori\dati acquisiti(3)\90_0_0.lvm','r');
files=[file1 file2 file3 file4 file5 file6];
%%
%parametri
g=9.81;
vettg=[0;0;g];
phi=0; %rotazione attorno all'asse z
theta=0; %rotazione attorno all'asse x
psi=0; %rotazione attorno all'asse y

theta=0;
R1=[1 0 0;0 cos(theta) sin(theta);0 -sin(theta) cos(theta)];
theta=pi;
R2=[1 0 0;0 cos(theta) sin(theta);0 -sin(theta) cos(theta)];

%2 rotazioni attorno y
psi=-pi/2;
R3=[cos(psi) 0 -sin(psi);0 1 0;sin(psi) 0 cos(psi)];
psi=pi/2;
R4=[cos(psi) 0 -sin(psi);0 1 0;sin(psi) 0 cos(psi)];

%2 rotazioni attorno a x
theta=-pi/2;
R5=[1 0 0;0 cos(theta) sin(theta);0 -sin(theta) cos(theta)];
theta=pi/2;
R6=[1 0 0;0 cos(theta) sin(theta);0 -sin(theta) cos(theta)];

%ogni colonna della matrice rotaz (:,k) avra' al suo interno il risultato
%di gT per la matrice di rotazione relativa al file
rotaz(:,1)=[R1*vettg;1];
rotaz(:,2)=[R2*vettg;1];
rotaz(:,3)=[R3*vettg;1];
rotaz(:,4)=[R4*vettg;1];
rotaz(:,5)=[R5*vettg;1];
rotaz(:,6)=[R6*vettg;1];
%%
%calcolo medie segnale
for i=1:6
    C=textscan(files(1,i), '%f32 %f32 %f32 %f32');
```

```

ay=C{2};
ax=C{3};
az=C{4};

aymedia=mean(ay);
axmedia=mean(ax);
azmedia=mean(az);

V(1,i)=axmedia;
V(2,i)=aymedia;
V(3,i)=azmedia;

fclose (files(1,i));
end
%%
%soluzione
soluzione=V*pinv(rotaz)
vx=[soluzione(1,1) soluzione(1,2) soluzione(1,3)];
vy=[soluzione(2,1) soluzione(2,2) soluzione(2,3)];
vz=[soluzione(3,1) soluzione(3,2) soluzione(3,3)];
Sx=sqrt(soluzione(1,1)^2+soluzione(1,2)^2+soluzione(1,3)^2)
Sy=sqrt(soluzione(2,1)^2+soluzione(2,2)^2+soluzione(2,3)^2)
Sz=sqrt(soluzione(3,1)^2+soluzione(3,2)^2+soluzione(3,3)^2)
Ox=soluzione(1,4)
Oy=soluzione(2,4)
Oz=soluzione(3,4)
versorex=vx/Sx
versorey=vy/Sy
versorez=vz/Sz

```

## VERIFICA TARATURA ACCELEROMETRO

```

%%
%Caricamento file
clear all;

file1 = fopen('C:\Documents and Settings\las.LAS-P\Desktop\Ste\nanotech new\lab
sensori\dati acquisiti(3)\-90_0_0.lvm','r');
file2 = fopen('C:\Documents and Settings\las.LAS-P\Desktop\Ste\nanotech new\lab
sensori\dati acquisiti(3)\90_0_0.lvm','r');
file3 = fopen('C:\Documents and Settings\las.LAS-P\Desktop\Ste\nanotech new\lab
sensori\dati acquisiti(3)\45_0_0.lvm','r');
file4 = fopen('C:\Documents and Settings\las.LAS-P\Desktop\Ste\nanotech new\lab
sensori\dati acquisiti(3)\-45_0_0.lvm','r');
file5 = fopen('C:\Documents and Settings\las.LAS-P\Desktop\Ste\nanotech new\lab
sensori\dati acquisiti(3)\0_-45_0.lvm','r');
file6 = fopen('C:\Documents and Settings\las.LAS-P\Desktop\Ste\nanotech new\lab
sensori\dati acquisiti(3)\0_45_0.lvm','r');
file7 = fopen('C:\Documents and Settings\las.LAS-P\Desktop\Ste\nanotech new\lab
sensori\dati acquisiti(3)\0_-90_0.lvm','r');
file8 = fopen('C:\Documents and Settings\las.LAS-P\Desktop\Ste\nanotech new\lab
sensori\dati acquisiti(3)\0_+90_0.lvm','r');
file9 = fopen('C:\Documents and Settings\las.LAS-P\Desktop\Ste\nanotech new\lab
sensori\dati acquisiti(3)\0_0_180.lvm','r');

%%
%Parametri
Sx =0.0733;
Sy =0.0736;
Sz = 0.0744;

```

```

Ox = 1.8996;
Oy = 1.8084;
Oz = 1.9336;
Offset=[Ox; Oy; Oz];
Sens=[Sx; Sy; Sz];
%matrici di rotazione
theta=-pi/2;
R1=[1 0 0;0 cos(theta) sin(theta);0 -sin(theta) cos(theta)];
theta=+pi/2;
R2=[1 0 0;0 cos(theta) sin(theta);0 -sin(theta) cos(theta)];
theta=+pi/4;
R3=[1 0 0;0 cos(theta) sin(theta);0 -sin(theta) cos(theta)];
theta=-pi/4;
R4=[1 0 0;0 cos(theta) sin(theta);0 -sin(theta) cos(theta)];
psi=-pi/4;
R5=[cos(psi) 0 -sin(psi);0 1 0;sin(psi) 0 cos(psi)];
psi=pi/4;
R6=[cos(psi) 0 -sin(psi);0 1 0;sin(psi) 0 cos(psi)];
psi=-pi/2;
R7=[cos(psi) 0 -sin(psi);0 1 0;sin(psi) 0 cos(psi)];
psi=+pi/2;
R8=[cos(psi) 0 -sin(psi);0 1 0;sin(psi) 0 cos(psi)];
theta=pi; %rotazione attorno all'asse x
R9=[1 0 0;0 cos(theta) sin(theta);0 -sin(theta) cos(theta)];

R=[R1 R2 R3 R4 R5 R6 R7 R8 R9];

numero_campioni=9;
files=[file1 file2 file3 file4 file5 file6 file7 file8 file9];
g=9.81;
vettg=[0;0;g];
versorex =[0.9997 -0.0089 0.0244];
versorey =[0.0171 0.9992 0.0367];
versorez =[-0.0095 0.0054 -0.9999];
M=[Sx.*versorex Ox;Sy.*versorey Oy;Sz.*versorez Oz];
%%
%Calcolo tensione media
for i=1:numero_campioni
    C=textscan(files(1,i), '%f32 %f32 %f32 %f32');
    giroscopio=C{1};
    ay=C{2};
    ax=C{3};
    az=C{4};

    aymedia=mean(ay);
    axmedia=mean(ax);
    azmedia=mean(az);

    Vm(1,i)=axmedia;
    Vm(2,i)=aymedia;
    Vm(3,i)=azmedia;

    fclose (files(1,i));
end
Vm
%%
%Calcolo tensione calcolata
k=1;
g1=R1*vettg;
vettg1=[g1;1];
Vc(:,k)=M*vettg1;

```



```

k=k+1;
g2=R2*vettg;
vettg2=[g2;1];
Vc(:,k)=M*vettg2;

```

```

k=k+1;
g3=R3*vettg;
vettg3=[g3;1];
Vc(:,k)=M*vettg3;

```

```

k=k+1;
g4=R4*vettg;
vettg4=[g4;1];
Vc(:,k)=M*vettg4;

```

```

k=k+1;
g5=R5*vettg;
vettg5=[g5;1];
Vc(:,k)=M*vettg5;

```

```

k=k+1;
g6=R6*vettg;
vettg6=[g6;1];
Vc(:,k)=M*vettg6;

```

```

k=k+1;
g7=R7*vettg;
vettg7=[g7;1];
Vc(:,k)=M*vettg7;

```

```

k=k+1;
g8=R8*vettg;
vettg8=[g8;1];
Vc(:,k)=M*vettg8;

```

```

k=k+1;
g9=R9*vettg;
vettg9=[g9;1];
Vc(:,k)=M*vettg9;

```

```

Vc

```

```

%%
%calcolo accelerazione misurata
offset_p=[Offset Offset Offset Offset Offset Offset Offset Offset Offset];
M1=M(:,1:3);
gmis=inv(M1)*(Vm-offset_p)

```

```

%%
%calcolo accelerazione calcolata
gcal=[vettg1 vettg2 vettg3 vettg4 vettg5 vettg6 vettg7 vettg8 vettg9]

```

```

%%
%Stima degli errori
Err=abs(Vm-Vc)
Err_acc=abs(gmis(1:3,:)-gcal(1:3,:))

```

## TARATURA GIROSCOPIO

```
%% Taratura giroscopio
% Caricamento file data
% i file sono senza header e composti di 4 colonne, di cui la prima
% rappresenta il giroscopio

clear;
file2 = load('C:\Documents and Settings\las.LAS-P\Desktop\Ste\nanotech new\lab
sensori\dati acquisiti\giroscopio90.lvm','r');
file3 = load('C:\Documents and Settings\las.LAS-P\Desktop\Ste\nanotech new\lab
sensori\dati acquisiti\giroscopiomeno90.lvm','r');
file4 = load('C:\Documents and Settings\las.LAS-P\Desktop\Ste\nanotech new\lab
sensori\dati acquisiti\giroscopio180.lvm','r');
file5 = load('C:\Documents and Settings\las.LAS-P\Desktop\Ste\nanotech new\lab
sensori\dati acquisiti\giroscopiomeno45.lvm','r');
file6 = load('C:\Documents and Settings\las.LAS-P\Desktop\Ste\nanotech new\lab
sensori\dati acquisiti\giroscopio45.lvm','r');
files=[file2 file3 file4 file5 file6];
numero_files=5;

theta=[90 -90 -180 -45 +45];
% theta=[-45];%lo definisco come una riga
freq = 1000;
% Implemento il filtro
% la freq di taglio e' fatta cosi' in qst filtro:
%rapporto tra frequenza alla quale vogliamo tagliare e frequenza di
%Nyquist, dove la freq di Nyquist e' appunto la freq di campionamento del
%segnale/2 (dal teo di shannon)
%quindi questo filtro e' un ordine due che taglia a 10 Hz una freq di 1000
%Hz
[z,p] = butter(2,10/(freq/2));
%G45f = filtfilt(z,p,G_0_0_45(:,1));
gir=1;%contatore per selezionare sempre la prima colonna quella del giroscopio
%%
%filtraggio
for i=1:numero_files%per ogni file

file_corrente= files(:,gir);
filtrati(:,i) = filtfilt(z,p,file_corrente(1:5000,1));

%%
%calcolo offset
% per prima cosa calcolo il punto di inizio curva
if i==0 % da togliere serviva se usavo anche il file1
    irinizio=1;
    iroffset=irinizio;
    irfine = 5000;
    offset(1,i)=mean(filtrati(1:iroffset,i))

else
derivata=diff(filtrati(:,i));
massimo=max(abs(derivata));
ir=0;
for ii=1:length(derivata)
    if ((abs(derivata(ii))>(0.18*massimo)) & ir==0)
        ir=ii;
        break
    end
end
end
```

```

irinizio=ir;

iroffset=irinizio-500;
offset(1,i)=mean(filtrati(1:iroffset,i));

%%
%calcolo del punto di fine curva
%qui voglio che la differenza sia piu' piccola di un numero piccolo
%considerando che ora sono prima del punto di massimo della curva devo
%prima scavalcarlo
ir=0;
for iii=length(derivata):(-1):1

    if ((abs(derivata(iii))>(0.11*massimo)) & ir==0)
        ir=iii;
        break
    end
end
irfine=ir;%parto leggermente prima del valore trovato per non utilizzare nel
calcolo dell
%offset parte della curva

%%
%calcolo sensibilità
%integro V-O su tutti i punti della curva
end%end dell'if del caso in cui lavoriamo sul primo file
%%
%integrazione
vmo =filtrati(irinizio:irfine,i)-offset(1,i);
% e integro usando trapz
W(:,i)=trapz(vmo)/freq;
gir=gir+4;
%end del for iniziale
end
O= mean (offset)
S= W*pinv(theta)

```

## VERIFICA PARAMETRI DI TARATURA GIROSCOPIO

```

clear all;

%%
%import file e dati
S =-0.0026;

vetoff =[2.2034      2.2049      2.2065      2.2043      2.2035];

offset =mean(vetoff);

file1 = load('C:\Documents and Settings\las.LAS-P\Desktop\Ste\nanotech new\lab
sensori\dati acquisiti\giroscopio90.lvm','r');
file2 = load('C:\Documents and Settings\las.LAS-P\Desktop\Ste\nanotech new\lab
sensori\dati acquisiti\giroscopiomeno90.lvm','r');
file3 = load('C:\Documents and Settings\las.LAS-P\Desktop\Ste\nanotech new\lab
sensori\dati acquisiti\giroscopio180.lvm','r');
file4 = load('C:\Documents and Settings\las.LAS-P\Desktop\Ste\nanotech new\lab
sensori\dati acquisiti\giroscopiomeno45.lvm','r');
file5 = load('C:\Documents and Settings\las.LAS-P\Desktop\Ste\nanotech new\lab
sensori\dati acquisiti\giroscopio45.lvm','r');
files=[file1 file2 file3 file4 file5];
numero_files=5;

```

```

theta=[90 -90 -180 -45 45];
freq = 1000;
% Implemento il filtro
[z,p] = butter(2,10/(freq/2));
gir=1;%contatore per selezionare sempre la prima colonna quella del giroscopio
figure(1)
hold on
colors=jet(numero_files);
integ(1:5000,1:numero_files)=0;

for i=1:numero_files%per ogni file
    file_corrente= files(:,gir);
    filtrati(:,i) = filtfilt(z,p,file_corrente(1:5000,1));
    %%
    % angolo calcolato
    for ii=2:length (filtrati(:,i))
        integ(ii,i)=integ(ii-1,i)+trapz(filtrati(ii-1:ii,i)-offset)/freq;
    end
    risultato(:,i)=integ(:,i)/S;
    tempo(:,i)=[1:length(filtrati(:,i))];
    max_risultato=max(abs(risultato(:,i)));
    %%
    %errore commesso. Viene calcolato sul massimo dell'integrale, quindi
    %dell'angolo percorso
    if (theta(i)>0)
        err(i)=abs(100*(max_risultato-theta(i))/(theta(i)))
    else
        err(i)=abs(100*(-max_risultato-theta(i))/(theta(i)))
    end
    %%
    %stampa dei dati

eval(sprintf('p%d=plot(tempo(:,%d),risultato(:,%d),'color',[colors(%d,:)],'LineWidth',2);',i,i,i,i,i));
    titolo=['Grafico delle ' int2str(i) ' tracce di controllo'];
    title(titolo);
    set(gca,'YTick',-225:45:225);
    xlabel('Tempo [ms]');
    ylabel('Angolo [°]');
    grid on;
    if i==1
        legends=sprintf('File 1 (%4.0f°)',theta(i));
        plots=p1;
    else
        legends=[legends; sprintf('File %d (%4.0f°)', i,theta(i))];
        plots=[plots;eval(sprintf('p%d',i))];
        legend(plots,legends,'Location','best');
    end
    gir=gir+4;
end

```

## VERIFICA DERIVA GIROSCOPIO

```

clear;
%%
%import file e dati
S =-0.0026;
vetoff =[2.2034      2.2049      2.2065      2.2043      2.2035];

```

```

offset =mean(vetoff);
file1=load('C:\Documents and Settings\las.LAS-P\Desktop\Ste\nanotech new\lab
sensori\dati acquisiti(3)\giroscopiomeno90,90,meno90,90_.txt','r');
files=[file1];
numero_files=1;
theta=[0];
freq=1000;
% Implemento il filtro
[z,p] = butter(2,10/(freq/2));
gir=1;
file_corrente= files(:,gir);
filtrati = filtfilt(z,p,file_corrente);
integ=0;
%%
%soluzione
for i=2:length(filtrati)
    integ(i)=integ(i-1)+trapz(filtrati(i-1:i)-offset)/freq;
end
risultato=integ/S;
tempo=[1:length(filtrati)];

%%
%calcolo errore
errore=100*(max(abs(risultato))-90)/90
errore_fine=max(risultato(7000:10000)-0)
%%
% stampa dei dati
figure(1)
plot (tempo,risultato);
set(gca,'YTick',-225:45:225);
xlabel('Tempo [ms]');
ylabel('Angolo [°]');
grid on
legends=sprintf('0° -90°');
legend(legends,'Location','best');
title ('verifica deriva 0° -90°');

```

## CALCOLO ANGOLI DI ROTAZIONE

### *Main program*

```

clear all
%acquisisco i dati e la frequenza di campionamento
file1 = load('C:\Documents and Settings\las.LAS-P\Desktop\Ste\nanotech new\lab
sensori\exercisel\exercise\esercizio7_13.lvm');
freq = 100;
%frequenza alla quale si vuole abbassare la frequenza del segnale
f=100;
%offset Fabio
offset_gir=1.6647;
%offset datasheet
offset_gir2=1.65;
so_acc=[0.0732    -0.0007    0.0018    1.8996;0.0013    0.0735    0.0027
1.8084;   -0.0007    0.0004    0.0744    1.9336];
%Ox = 1.8996;
vettg=[0 0 9.81 1];
lambda = [0.003 0.005 0.01];
%sens Fabio
sensgir=-0.0033;
%sens datasheet

```

```

sensgir2=-0.00323;
% Rapporto di frequenza
rf=freq/f;
istanti=3000;

% Riduzione di frequenza del segnale
istantirf=floor(istanti/rf);

for ii=1:istantirf
lento(ii,1)=file1(ii*rf,1);
lento(ii,2)=file1(ii*rf,2);
lento(ii,3)=file1(ii*rf,3);
lento(ii,4)=file1(ii*rf,4);
end
lento_lenght= length(lento);
[z,p] = butter(2,10/(freq/2));
%qui ho messo solo fino a 10000 xke se no non finivo piu...
%filtrato = filtfilt(z,p,file1(1:10000,:));
dati_filtrati = filtfilt(z,p,lento(:, :));

for i=1:length(lambda)
    %risultato dati Fabio
    [risultato(i,:),angolo_gir(i,:)] = calcola_angolo (dati_filtrati,
offset_gir, sensgir, f, so_acc,lambda(i));
    %stampa grafici(risultato, 'Dati Fabio lambda=',1);
    %risultato dati datasheet
    [risultato2(i,:),angolo_gir2(i,:)] = calcola_angolo (dati_filtrati,
offset_gir2, sensgir2, f, so_acc,lambda(i));

end
stampa_grafici(risultato,angolo_gir,'Angoli di rotazione | Dati
Calcolati',1,lambda);
stampa_grafici(risultato2,angolo_gir2, 'Angoli di rotazione | Dati
Datasheet',2,lambda);

```

### Funzione Calcola\_angolo

```

function [risultato, angolo_gir] = calcola_angolo (dati_filtrati, offset_gir,
sensgir, f, so_acc, lambda)
    %input:
    %dati_filtrati deve essere una matrice a 4 colonne con giroscopio ax ay az
    %so_acc e' la matrice contentente sensibilita' e offset dell'accelerometro
    %f = frequenza alla quale stiamo lavorando, anche se ridotta
    %output:
    %risultato = angolo corretto
    %angolo_gir = angolo calcolato dal giroscopio prima della correzione

    filtrato_length = length(dati_filtrati(:,1));
    for ii=1:filtrato_length
        V(1:3,ii)=[dati_filtrati(ii,2);dati_filtrati(ii,3);dati_filtrati(ii,4)];
        %calcolo le accelerazioni misurate dall'accelerometro
        %qui ho bisogno di un vettore
        %il vettore delle accelerazioni in ogni punto (ax;ay;az)=
        %(V-O)/S: quella e' la divisione al contrario
        %cmq e' come dire (V-O)/S
    end

```

```

acc(1:3,ii)=inv(so_acc(1:3,1:3))\ (V(:,ii)-so_acc(1:3,4));
modulo=sqrt(acc(1,ii)^2+acc(2,ii)^2+acc(3,ii)^2);
%Calcolo del versore dall'accelerometro
%a questo punto e' come dividere tutte le componenti di un vettore per
il suo
%modulo e trovo i tre vettori di base, i versori
versori(1:3,ii)=acc(1:3,ii)/modulo;
%Calcolo angolo dall'accelerometro
%tramite la formula angolo=arccoseno(-accelerazione/g) e la porto in
gradi
%dai radianti e non c'e' g perche in pratica a/g e' proprio il versore
% che ho gia calcolato. Cmq questo e' ovviamente l'angolo
% rispetto all'asse X esclusivamente!
angolo(ii)=(180/pi)*asin(versori(1,ii));
end

%         figure(4)
%         plot (angolo);
%         title('angolo acc')
risultato=1;
integ=0;
for i=2:filtrato_length
    integ(i)=integ(i-1)+trapz(dati_filtrati(i-1:i,1)-offset_gir)/f;
end
angolo_gir=integ/sensgir;
%         figure(5)
%         plot (angolo_gir);
%         title('angolo giroscopio')
vers_gir= calcolo_versore (dati_filtrati, angolo_gir, so_acc);

for ii=1:filtrato_length

p_vett(ii)=(dot(vers_gir(1:3,ii),versori(1:3,ii)))/((norm(vers_gir(1:3,ii))*(norm(versori(1:3,ii)))));
    delta(ii)=acos(p_vett(ii));
end

integ(1:filtrato_length)=0;
for ii=2:filtrato_length
    omega_corr(ii-1)=lambda*delta(ii-1);
    omega_ok(ii)=dati_filtrati(ii,1)+omega_corr(ii-1);
    integ(ii)=integ(ii-1)+trapz(omega_ok(ii-1:ii)-offset_gir)/f;

end
risultato=integ/sensgir;
end

```

### **Funzione calcola\_versore**

```

function [vers_gir] = calcolo_versore (dati_filtrati, angolo_gir, so_acc)
    dati_filtrati_length=length(dati_filtrati);
    for ii=1:dati_filtrati_length
        if (dati_filtrati(1,2)>so_acc(1,4))&&(dati_filtrati
(1,2)>dati_filtrati(1,3))
            %il valore dell'asse x e' maggiore di quello di y ed e' positivo->
l'asse
                %x punta verso il basso perche sente -g
                %ora stabiliamo il verso di rotazione
                if ((dati_filtrati(51,3)-dati_filtrati(1,3))/50)>0

```

```

        %rotazione negativa (y verso x)
        %versore giroscopio punto per punto in radianti
        %
        a=1
        vers_gir(1:3,ii) = [-cos(angolo_gir(ii)*pi/180);
+sin(angolo_gir(ii)*pi/180); 0];
        %vers_gir(1:3,ii) = [-sin(angolo_gir(ii)*pi/180);
+cos(angolo_gir(ii)*pi/180); 0];

        else
            %rotazione pos
            vers_gir(1:3,ii) = [-cos(angolo_gir(ii)*pi/180); -
sin(angolo_gir(ii)*pi/180); 0];
            %
            a=2

        end

        elseif
            (dati_filtrati(1,3)>so_acc(2,4))&&(dati_filtrati(1,3)>dati_filtrati(1,2))
                %asse y rivolto verso il basso
                if (dati_filtrati(51,2)-dati_filtrati(1,2))/50>0
                    vers_gir(1:3,ii) = [sin(angolo_gir(ii)*pi/180); -
cos(angolo_gir(ii)*pi/180); 0];
                    %
                    a=3
                else
                    vers_gir(1:3,ii) = [-sin(angolo_gir(ii)*pi/180); -
cos(angolo_gir(ii)*pi/180);0];
                    %
                    a=4
                end

            elseif
                (dati_filtrati(1,2)<so_acc(1,4))&&(dati_filtrati(1,2)<dati_filtrati(1,3)) %asse
x rivolto verso l'alto
                if (dati_filtrati(51,3)-dati_filtrati(1,3))/50>0
                    %
                    a=5
                    vers_gir(1:3,ii) = [cos(angolo_gir(ii)*pi/180);
sin(angolo_gir(ii)*pi/180); 0];
                else
                    %
                    a=6
                    vers_gir(1:3,ii) = [cos(angolo_gir(ii)*pi/180); -
sin(angolo_gir(ii)*pi/180);0];
                end

            elseif
                (dati_filtrati(1,3)>so_acc(2,4))&&(dati_filtrati(1,3)>dati_filtrati(1,2)) %asse
y rivolto verso l'alto
                if (dati_filtrati(51,2)-dati_filtrati(1,2))/50>0
                    %
                    a=7
                    vers_gir(1:3,ii) = [sin(angolo_gir(ii)*pi/180);
cos(angolo_gir(ii)*pi/180); 0];
                else
                    %
                    a=8
                    vers_gir(1:3,ii) = [-sin(angolo_gir(ii)*pi/180);
cos(angolo_gir(ii)*pi/180);0];
                end

            else
                errore = 'errore nel calcolo del versore'
            end

        end
    end
end

```



### *Funzione stampa\_grafici*

```
function [] = stampa_grafici (M,angolo_gir, titolo,num_figura,lambda)
    M=M';
    angolo_gir=angolo_gir';
    figure(num_figura)
    tempo=[1:length(M(:,1))];
    p=plot (tempo/100,M(:,1),tempo/100,M(:,2),tempo/100, M(:,3), tempo/100,
angolo_gir(:,1));
    title(titolo)
    set(gca,'YTick',-225:45:225);
    xlabel('Tempo [s]');
    ylabel('Angolo [°]');
    grid on
    legends=sprintf('lambda= (%5.4f) ',lambda(1));
    legends=[legends; sprintf('lambda= (%5.4f) ',lambda(2))];
    legends=[legends; sprintf('lambda= (%5.4f) ',lambda(3))];
    %qui sono necessari gli spazi perche' all'interno della matrice legends
    %tutti gli elementi devono avere (ovviamente) la stessa lunghezza
    %cmq alla stampa del grafico non saranno visibili
    legends=[legends; sprintf('giroscopio      ')];
    legend(p,legends,'Location','best');
end
```

## Bibliografia

- Datasheet accelerometro LIS3L02AS4  
<http://www.st.com/stonline/products/literature/od/10221/lis3l02as4.pdf>
- Datasheet giroscopio LISY300AL  
<http://www.st.com/stonline/products/literature/ds/14753/lisy300al.pdf>
- <http://en.wikipedia.org/wiki/Accelerometer>
- <http://it.wikipedia.org/wiki/Accelerometro>
- I SENSORI INERZIALI DI BASSO COSTO PER LA NAVIGAZIONE GEODETICA - Mattia De Agostino - Politecnico di Torino, Dipartimento di Ingegneria dell'Ambiente, del Territorio e delle Geotecnologie (DITAG)
- Dispense "Il giroscopio Mems" – A. Longoni – Politecnico di Milano 2009-2010
- MICROSYSTEM DESIGN – Stephen D. Senturia - KLUWER ACADEMIC PUBLISHERS
- Appunti e dispense – Stefano Rossi