



**UNIVERSITA' POLITECNICA DELLE  
MARCHE  
FACOLTA' DI INGEGNERIA**

---

Corso di Laurea triennale: **Ingegneria informatica e dell'automazione**

**Sviluppo di algoritmi di data fusion per  
bussole elettroniche e piattaforme inerziali**

**Development of data fusion algorithms for  
electronic compasses and inertial platforms**

Relatore: Chiar.mo  
Prof. **Paolo Crippa**

Tesi di Laurea di:  
**Mattia Vesperini**

Correlatore: Chiar.mo  
Prof. **Giorgio Biagetti**

**A.A. 2018/ 2019**



# INDICE

INTRODUZIONE .....	5
CAPITOLO 1: MATLAB .....	6
1.1 CHE COS'È .....	6
1.2 COME L'HO UTILIZZATO .....	7
CAPITOLO 2: PIATTAFORME INERZIALI .....	8
2.1 COSA SONO .....	8
2.2 ACCELEROMETRO: .....	8
2.3 GIROSCOPIO:.....	9
2.4 MAGNETOMETRO:.....	12
CAPITOLO 3: ALGORITMI E DATI DI PARTENZA.....	13
3.1 ACQUISIZIONE DESTRA SINISTRA E RITORNO:.....	14
CAPITOLO4: ALGORITMO E-COMPASS MATLAB .....	20
4.1 CHE COS'È .....	20
4.2 QUATERNIONI E ANGOLO DI EULERO.....	20
4.2 ANGOLI DI EULERO.....	22
4.3 FUNZIONAMENTO ALGORITMO.....	23
CAPITOLO 5: DATA FUSION .....	29
5.1 COSA SIGNIFICA .....	29
5.2 IL MIO ALGORITMO DI DATA FUSION.....	29
5.3 RISULTATI.....	35
CONCLUSIONI:.....	39
BIBLIOGRAFIA .....	41
SITOGRAFIA .....	41



## **INTRODUZIONE**

Lo scopo del tirocinio sul quale si basa la tesi è stato quello di analizzare dei dati presi mediante diversi sensori (magnetometro, giroscopio, accelerometro) e studiare il loro comportamento in diverse situazioni; come ad esempio una camminata di 30 minuti intorno alla facoltà di ingegneria dell'Università Politecnica delle Marche e su una semplice rotazione di 90 gradi a destra e a sinistra. Analizzando e confrontando i risultati ottenuti, i segnali raccolti erano molto rumorosi e non potevano essere utilizzati per una corretta analisi e stima della posizione nel tempo.

Effettuando un'esplorazione algoritmica si è scelto di utilizzare un nuovo algoritmo Matlab "E-compass" che consente già una buona stima e sul quale ci si è focalizzati particolarmente vista l'implementazione ancora poco usata e diffusa.

Dopodiché si è applicato un filtro di Kalman per cercare di avere un risultato ancora meno rumoroso e più preciso possibile tramite un efficiente algoritmo di data fusion dopo aver notato che il solo utilizzo dell'algoritmo E-compass non consentiva una corretta stima dell'orientazione nel tempo.

# **CAPITOLO 1: MATLAB**

## **1.1 CHE COS'È**

Il software utilizzato per l'elaborazione dei dati è MATLAB (abbreviazione di Matrix Laboratory) che è un ambiente per il calcolo numerico e l'analisi numerica scritto in C, comprendente il linguaggio di programmazione ideato dalla MathWorks. MATLAB consente di costruire matrici, visualizzare funzioni e dati, implementare algoritmi, creare interfacce front-end, e interfacciarsi con altri tipi di programmi. Tale particolarità rende questo software uno strumento particolarmente versatile ed efficiente per l'elaborazione dei dati e lo sviluppo di modelli. Infatti, MATLAB permette di trattare grandi insiemi di dati e di effettuare calcoli molto complessi tramite un numero ridotto di righe di codice. È usato da milioni di persone nell'industria e nelle università per via dei suoi numerosi strumenti a supporto dei più svariati campi di studio applicati e funziona su diversi sistemi operativi tra cui Windows, MacOS, GNU/Linux e Unix e con le ultime versioni anche mobile su Android e iOS.

## **1.2 COME L'HO UTILIZZATO**

Per uso universitario avevo già preso conoscenza con il programma sia per la soluzione di semplici sistemi matematici e successivamente anche nell'elaborazione dei segnali in materie come elettrotecnica in cui in determinati analisi circuitali dovevamo controllare il corretto funzionamento del circuito tramite uno script da noi perfezionato. Il tutto è stato reso possibile tramite la Università Politecnica delle Marche che ha acquistato la licenza accademica.

## **CAPITOLO 2: PIATTAFORME INERZIALI**

### **2.1 COSA SONO**

Un'unità di misura inerziale o piattaforma inerziale (abbreviazione di inertial measurement unit, o IMU) è un sistema elettronico che si basa su sensori inerziali, come gli accelerometri, magnetometri, giroscopi, che permettono un monitoraggio della dinamica di un mezzo in movimento, il quale può essere utilizzato dal computer o centralina di bordo per attuare eventuali correttivi. Sono anche detti sensori di tipo MEMS (Micro Electro-Mechanical Systems): sono sensori micrometrici, che permettono di trasformare un input meccanico come la pressione, l'accelerazione in un semplice impulso elettrico, quantificabile e misurabile da una centralina.

### **2.2 ACCELEROMETRO:**

Strumento di misura in grado di rilevare e/o misurare l'accelerazione, effettuando il calcolo della forza rilevata rispetto alla massa dell'oggetto (forza per unità di massa).

Per avere uno schema mentale che ci aiuti a visualizzare in qualche modo come opera un accelerometro, immaginiamolo come costituito da una sfera al centro di un cubo, sospesa da 3 molle che la attraversano, agganciate a loro volta al centro di ogni faccia del cubo:

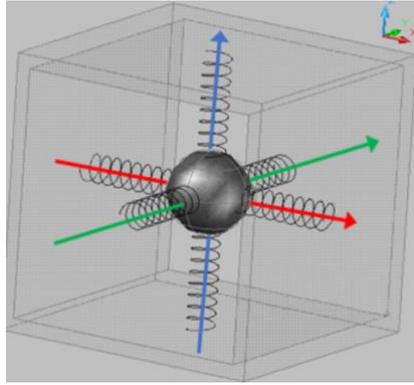


Figura 1:esempio accelerometro

Muovendo il cubo nello spazio, la sfera si muoverà al suo interno allungando e comprimendo, di conseguenza, le molle che la tengono sospesa. La misurazione del grado di compressione delle molle permette di stabilire se c'è stata un'accelerazione nella direzione in cui si trova la molla compressa e quindi anche di quantificarla.

### **2.3 GIROSCOPIO:**

Misura la velocità di rotazione sugli assi X,Y,Z calcolate tramite masse microscopiche, la cui loro dimensione oscilla tra 1 e 100 micrometri, che si muovono in base ai cambiamenti della velocità angolare. Il giroscopio è un dispositivo fisico in grado di muoversi in modo peculiare sul proprio asse. Il principio su cui si basa il giroscopio è quello della legge di conservazione del momento angolare, secondo cui il momento angolare di un sistema rimane costante a meno che agisca su di esso una forza esterna. Quando questo accade il corpo tende a mantenere un orientamento fisso, con l'asse che punta nella direzione della rotazione.

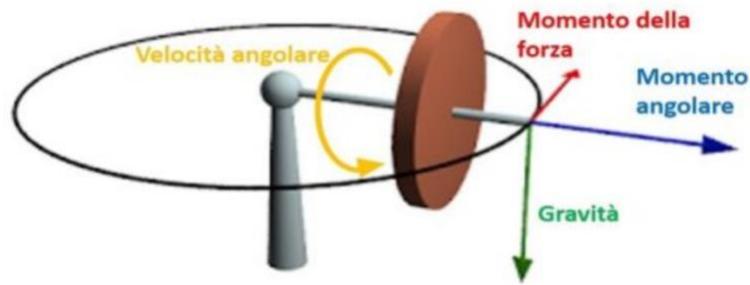


Figura 2: Funzionamento Giroscopio

Nel disegno, il disco marrone rappresenta un corpo in grado di muoversi intorno al proprio asse, al quale può essere applicata una forza di rotazione, indicata con la freccia rossa, che nello specifico descrive il momento della forza. Quando il disco comincia a ruotare in senso antiorario, come indicato dalla freccia gialla della velocità angolare (rappresentata con il simbolo  $\omega$ ) si crea un momento angolare (spesso abbreviato con la lettera  $L$ ) che può essere misurato secondo la formula:

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

Ossia il prodotto tra la velocità angolare ( $\omega$ ) e il momento di inerzia ( $I$ ), che misura l'inerzia del corpo al mutare della sua velocità angolare. [9]

Come affermato in precedenza, il giroscopio misura la velocità angolare; a livello fisico avremo quindi:

$$\dot{\theta} = \frac{d\theta}{dt}$$

ossia la derivata della posizione angolare sul tempo. Per calcolare la posizione angolare in modo continuo bisogna effettuare questa integrazione:

$$\int_0^t \dot{\Theta}(t) dt = \Theta(t)$$

Il problema dei sistemi digitali è l'impossibilità di effettuare un'integrazione continua, pertanto va effettuata una sommatoria di un numero finito di velocità angolari calcolate entro un intervallo temporale costante  $T_s$ , da cui:

$$\theta(t) = \int_0^t \dot{\theta}(t) dt \approx \sum_0^t \dot{\theta}(t) T_s$$

Ovviamente l'approssimazione introduce errori poiché se i valori letti dal giroscopio cambiassero più velocemente dell'intervallo temporale costante, non saranno individuabili e quindi risulterà un errore nella rilevazione finale dell'angolo. Questo errore è definito drift.

## **2.4 MAGNETOMETRO:**

Strumento di misura per la forza e talvolta la direzione dei campi magnetici, compresi quelli sulla o vicino alla Terra e nello spazio. I magnetometri vengono anche utilizzati per calibrare elettromagneti e magneti permanenti e per determinare la magnetizzazione dei materiali. I magnetometri utilizzati specificamente per misurare il campo terrestre sono di due tipi: assoluti e relativi (classificati in base ai loro metodi di calibrazione). I magnetometri assoluti sono calibrati con riferimento alle loro costanti interne note. I magnetometri relativi devono essere calibrati facendo riferimento a un campo magnetico noto e misurato con precisione. Il più semplice magnetometro assoluto, ideato da CF Gauss nel 1832, è costituito da un magnete a barra permanente sospeso orizzontalmente da una fibra d'oro. Misurare il periodo di oscillazione del magnete nel campo magnetico terrestre fornisce una misura dell'intensità del campo. Il magnetometro è molto influenzato dai campi magnetici che lo circondano.

## CAPITOLO 3: ALGORITMI E DATI DI PARTENZA

Lo studio del comportamento nel tempo e nello spazio di un oggetto tramite le piattaforme inerziali significa analizzare i risultati di diverse prove sotto indicate tramite l'analisi di grafici che simulano l'andamento di un qualsiasi oggetto dotato di determinati sensori.

Le prove prese in considerazione tra le disponibili si basano su due algoritmi sviluppati in Matlab: uno più semplice dal nome "Acquisizione destra, sinistra e ritorno" e uno più articolato dal nome "Acquisizione 30 minuti". Entrambi hanno in input come file ".txt" i dati appartenenti a magnetometro, giroscopio e accelerometro. Ogni file ".txt" ha la seguente composizione:

- Prima colonna: timeStamp (ordine di stampa)
- Seconda colonna: Orario
- Terza colonna: Asse delle X
- Quarta colonna: Asse delle Y
- Quinta colonna: Asse delle Z

```
21767 084329.582649 0.7744140625 0.1103515625 0.609375
21773 084329.627698 0.7744140625 0.1103515625 0.6083984375
21779 084329.673416 0.7744140625 0.1103515625 0.6083984375
21786 084329.732617 0.7744140625 0.109375 0.607421875
21792 084329.777711 0.7734375 0.1103515625 0.6083984375
21798 084329.823416 0.7744140625 0.111328125 0.607421875
21804 084329.882633 0.7744140625 0.109375 0.609375
21811 084329.928322 0.7744140625 0.1103515625 0.607421875
```

Figura 3: Esempio file Accelerometro.txt

### 3.1 ACQUISIZIONE DESTRA SINISTRA E RITORNO:

Con Acquisizione Destra Sinistra e Ritorno si intende un algoritmo che analizza un movimento elementare di un oggetto che va prima a destra, poi a sinistra e infine alla posizione di partenza. Non ci sarà mai quindi una variazione angolare superiore ai 90°. I risultati ottenuti sono rappresentati dal seguente output che combina insieme le analisi mediante giroscopio, accelerometro e magnetometro:

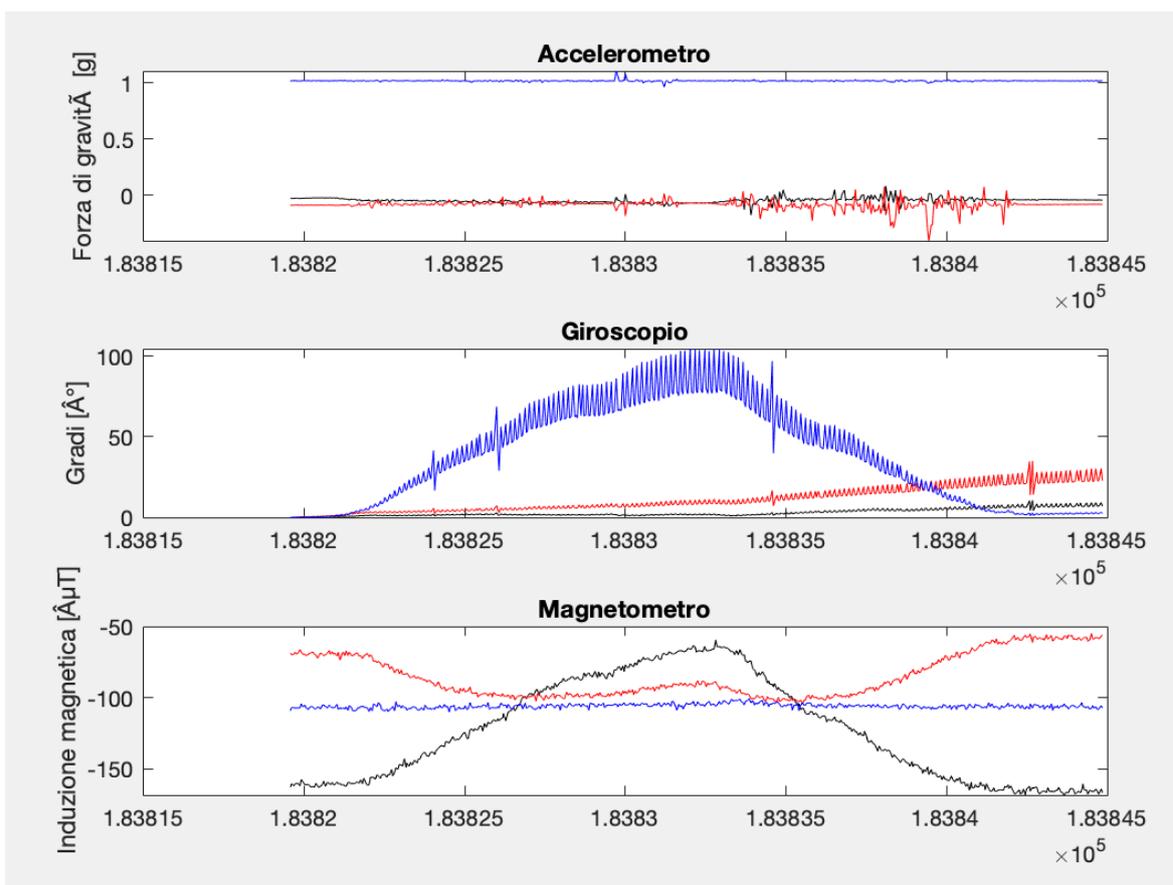


Figura 4: Asse X(nero), Asse Y(rosso), Asse Z(blu)

Il codice che genera questo output è il seguente:

```
clc
close all
clear all
format long
acc_matrix = importdata ("Accelerometro (Rotazione sx-dx-sx).txt");
timestamp = acc_matrix (:,1);           %uguale in tutti i 3 file
orario = acc_matrix (:,2);             %uguale in tutti i 3 file
acc_x_axis = acc_matrix (:,3);
acc_y_axis = acc_matrix (:,4);
acc_z_axis = acc_matrix (:,5);
deltaT = [0 ; diff(orario)];           %differenza tra le notifiche ricevute in microsecondi
figure('Name','Accelerometro','NumberTitle','off');
plot (orario, acc_x_axis, 'black', orario, acc_y_axis, 'red', orario, acc_z_axis, 'blue')
xlabel('Tempo [hhmmss.micros]')
ylabel('Forza di gravità [g]')
title('Accelerometro')

gyr_matrix = importdata ("Giroscopio (Rotazione sx-dx-sx).txt");
gyr_x_axis = gyr_matrix (:,3);
gyr_y_axis = gyr_matrix (:,4);
gyr_z_axis = gyr_matrix (:,5);
integrale_x = cumtrapz (gyr_x_axis);
integrale_y = cumtrapz (gyr_y_axis);
integrale_z = cumtrapz (gyr_z_axis);
gradi_x_axis = integrale_x .* deltaT;
gradi_y_axis = integrale_y .* deltaT;
gradi_z_axis = integrale_z .* deltaT;
figure('Name','Giroscopio','NumberTitle','off');
plot (orario, gradi_x_axis, 'black', orario, gradi_y_axis, 'red', orario, gradi_z_axis, 'blue')
xlabel('Tempo [hhmmss.micros]')
ylabel('Gradi [°]')
title('Giroscopio')

%xtickformat('hh-mm-ss.SSSSSS')

magn_matrix = importdata ("Magnetometro (Rotazione sx-dx-sx).txt");
magn_x_axis = magn_matrix (:,3);
magn_y_axis = magn_matrix (:,4);
magn_z_axis = magn_matrix (:,5);
figure('Name','Magnetometro','NumberTitle','off');
plot (orario, magn_x_axis, 'black', orario, magn_y_axis, 'red', orario, magn_z_axis, 'blue')
xlabel('Tempo [hhmmss.micros]')
ylabel('Induzione magnetica [ÅμT]')
title('Magnetometro')

%***** confronto tra i 3 sensori *****
figure('Name','Confronto tra i 3 sensori','NumberTitle','off');
ax1 = subplot(3,1,1);
plot (ax1, orario, acc_x_axis, 'black', orario, acc_y_axis, 'red', orario, acc_z_axis, 'blue')
title(ax1,'Accelerometro')
ylabel(ax1,'Forza di gravità [g]')

ax2 = subplot(3,1,2);
%figure('Name','Confronto giroscopio e magnetometro','NumberTitle','off');
plot (ax2, orario, gradi_x_axis, 'black', orario, gradi_y_axis, 'red', orario, gradi_z_axis, 'blue')
title(ax2,'Giroscopio')
ylabel(ax2,'Gradi [°]')

ax3 = subplot(3,1,3);
plot(ax3, orario, magn_x_axis, 'black', orario, magn_y_axis, 'red', orario, magn_z_axis, 'blue')
title(ax3,'Magnetometro')
ylabel(ax3,'Induzione magnetica [ÅμT]')
```

Figura 5: Codice "Rotazione\_dx\_sx.m"

Nonostante l'insieme dei segnali raccolti sia molto elementare e di breve durata, si nota che comunque l'algoritmo restituisce in output un segnale molto rumoroso, soprattutto nell'analisi del giroscopio. Si evince quindi che considerando un segnale più complesso e di durata maggiore si possono avere delle criticità importanti.

## 3.2 ACQUISIZIONE 30 MINUTI:

Con “acquisizione 30 minuti” si fa riferimento a un algoritmo che si basa sull’analisi di più segnali ottenuti durante un percorso di 30 minuti intorno al complesso universitario di Monte Dago (Ancona). I sensori erano piazzati all’interno di uno zaino, non perfettamente stabili, trasportato a passo più omogeneo possibile. Il percorso è composto da dislivelli e da andamenti non uniformi, aggravanti che ne hanno influenzato l’analisi del segnale tramite l’algoritmo.



Figura 6: Percorso e rappresentazione grafica dell’andamento.

In questo caso il codice dell'algorithmo adoperato è il seguente:

```

clc
close all
clear all
format long
acc_matrix = importdata ("Accelerometro.txt");
timestamp = acc_matrix (:,1);           %uguale in tutti i 3 file
orario = acc_matrix (:,2);             %uguale in tutti i 3 file
acc_x_axis = acc_matrix (:,3);
acc_y_axis = acc_matrix (:,4);
acc_z_axis = acc_matrix (:,5);
deltaT = [0 ; diff(orario)];           %differenza "temporale" tra le notifiche
%il vettore "orario" Ã  espresso in hhmss.micross quindi ad ogni minuto
%successivo vi Ã  un aumento fittizio di 40 secondi e ad ogni ora successiva
%vi Ã  un aumento fittizio di 40 minuti
for i = 1:length(deltaT)
    if (deltaT(i)>1 && deltaT(i)<41)
        deltaT(i) = deltaT(i) - 40;
    end
    if (deltaT (i)>41)
        deltaT(i) = deltaT(i) - 4040;
    end
end
%ottengo un vettore dato dalla somma dei delta T in modo da iniziare a
%graficare il tempo da zero
tempo = cumsum (deltaT);               %gli elementi di "tempo" sono espressi in secondi.microsecondi
figure('Name','Accelerometro','NumberTitle','off');
plot (tempo, acc_x_axis, 'black', tempo, acc_y_axis, 'red', tempo, acc_z_axis, 'blue')
xlabel('Tempo [s]')
ylabel('Forza di gravitÃ  [g]')
legend('Acc_x','Acc_y','Acc_z','Giroscopio');
gyr_matrix = importdata ("Giroscopio.txt");
gyr_x_axis = gyr_matrix (:,3);
gyr_y_axis = gyr_matrix (:,4);
gyr_z_axis = gyr_matrix (:,5);
integrale_x = cumtrapz (gyr_x_axis);
integrale_y = cumtrapz (gyr_y_axis);
integrale_z = cumtrapz (gyr_z_axis);
gradi_x_axis = integrale_x .* deltaT;
gradi_y_axis = integrale_y .* deltaT;
gradi_z_axis = integrale_z .* deltaT;
%ciclo per ottenere i gradi compresi tra -360Ã  e 360Ã
for i = 1:length(gradi_x_axis)
    while (abs(gradi_x_axis(i))>360)
        if (gradi_x_axis(i)>0)
            gradi_x_axis (i) = gradi_x_axis(i) - 360;
        end
        if (gradi_x_axis(i)<=-360)
            gradi_x_axis (i) = gradi_x_axis(i) + 360;
        end
    end
    while (abs(gradi_y_axis(i))>360)
        if (gradi_y_axis(i)>0)
            gradi_y_axis(i) = gradi_y_axis(i) - 360;
        end
        if (gradi_y_axis(i)<=-360)
            gradi_y_axis(i) = gradi_y_axis(i) + 360;
        end
    end
    while (abs(gradi_z_axis(i))>360)
        if (gradi_z_axis(i)>0)
            gradi_z_axis(i) = gradi_z_axis(i) - 360;
        end
    end
end

```

```

        if (gradi_z_axis(i)<=-360)
            gradi_z_axis(i) = gradi_z_axis(i) + 360;
        end
    end
end
figure('Name','Giroscopio','NumberTitle','off');
plot (tempo, gradi_x_axis, 'black', tempo, gradi_y_axis, 'red', tempo, gradi_z_axis, 'blue')
xlabel('Tempo [s]')
ylabel('Gradi [Å°]')
legend ('Asse x', 'Asse y', 'Asse z')
title('Giroscopio')
magn_matrix = importdata ("Magnetometro.txt");
magn_x_axis = magn_matrix (:,3);
magn_y_axis = magn_matrix (:,4);
magn_z_axis = magn_matrix (:,5);
figure('Name','Magnetometro','NumberTitle','off');
plot (tempo, magn_x_axis, 'black', tempo, magn_y_axis, 'red', tempo, magn_z_axis, 'blue')
xlabel('Tempo [s]')
ylabel('Induzione magnetica [ÅµT]')
legend ('Asse x', 'Asse y', 'Asse z')
title('Magnetometro')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure('Name','Confronto tra i 3 sensori','NumberTitle','off');
ax1 = subplot(3,1,1);
plot (ax1, tempo, acc_x_axis, 'black', tempo, acc_y_axis, 'red', tempo, acc_z_axis, 'blue')
legend ('Asse x', 'Asse y', 'Asse z')
title(ax1,'Accelerometro')
ylabel(ax1,'Forza di gravitÅ [g]')
ax2 = subplot(3,1,2);
plot (ax2, tempo, gradi_x_axis, 'black', tempo, gradi_y_axis, 'red', tempo, gradi_z_axis, 'blue')
legend ('Asse x', 'Asse y', 'Asse z')
title(ax2,'Giroscopio')
ylabel(ax2,'Gradi [Å°]')
ax3 = subplot(3,1,3);
plot(ax3, tempo, magn_x_axis, 'black', tempo, magn_y_axis, 'red', tempo, magn_z_axis, 'blue')
legend ('Asse x', 'Asse y', 'Asse z')
title(ax3,'Magnetometro')
ylabel(ax3,'Induzione magnetica [ÅµT]')

```

Che genera un output del tipo:

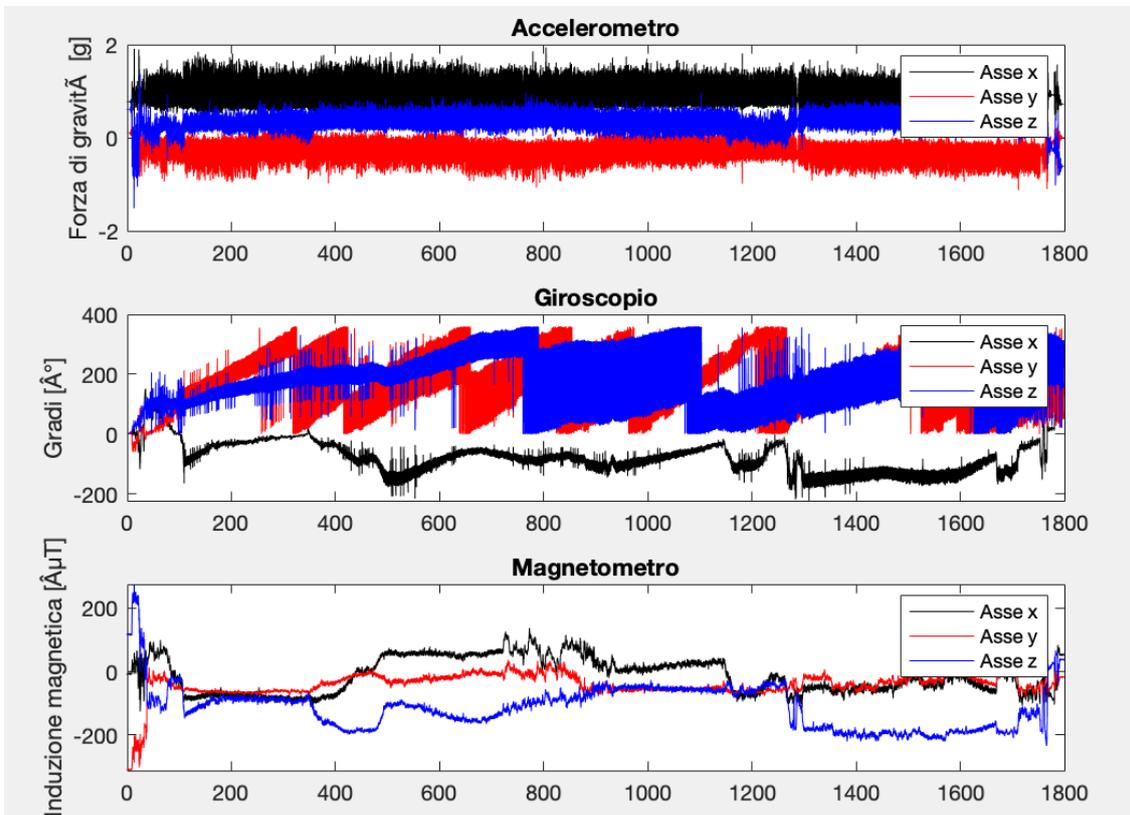


Figura 7: Output Acquisizione 30 minuti.

Come si evince dai grafici il rumore presente nei segnali è talmente elevato da rendere impossibile la stima della posizione dell'oggetto. Risulta nulla o poco utile l'analisi appena effettuata.

## **CAPITOLO4: ALGORITMO E-COMPASS MATLAB**

### **4.1 CHE COS'È**

È un algoritmo di fusion che riceve in input i dati presi dall'accelerometro e dal magnetometro. Restituisce in output dei quaternioni che stimano in modo molto efficace l'orientazione di un corpo nello spazio. Usa l'intensità del campo magnetico noto e la corretta accelerazione di un dispositivo puntato verso "nord vero" per determinare la declinazione magnetica. Per nord vero si intende la distanza angolare tra Nord Geografico e il Nord Magnetico e il valore della declinazione magnetica varia da luogo a luogo e cambia anche nel tempo perché il Nord Magnetico a differenza di quello Geografico non è statico.

Per la corretta stima della posizione si trasformano i risultati da quaternioni in angolo Eulero.

### **4.2 QUATERNIONI E ANGOLO DI EULERO**

#### **QUATERNIONI:**

Una volta studiati i numeri complessi ci si chiede se è possibile estendere questa struttura algebrica in qualcosa di più generale.

Nel Cinquecento i matematici italiani come Tartaglia, Bombelli scoprirono questa struttura attraverso lo studio delle equazioni di

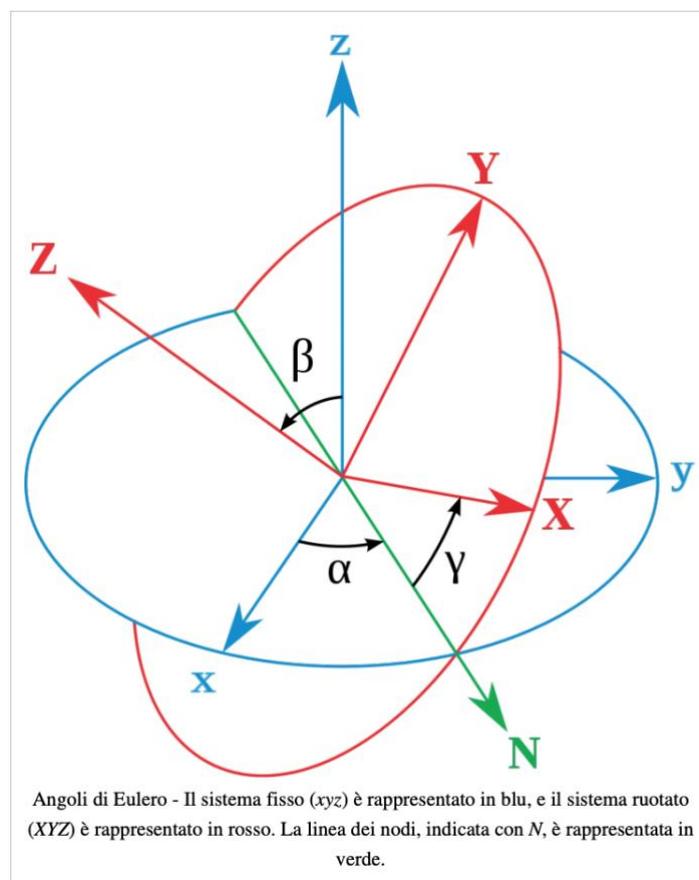
terzo grado. Facendo dei tentativi per la loro risoluzione arrivarono a concepire che esisteva una legge, dall'equazione " $i^2 = -1$ ", sulla quale si basa la risoluzione dei numeri complessi.

Con Gauss si riuscì ad interpretare in un modo straordinario la geometria dei numeri complessi e lo si fece introducendo il piano di Gauss, cioè un piano che descrive la geometria di un numero complesso attraverso una corrispondenza con delle coppie di numeri reali. È possibile estendere questo insieme di numeri? La risposta a questa domanda è data da William Roman Hamilton che studiò come l'evoluzione di un sistema a due dimensioni non è un sistema a tre dimensioni bensì uno spazio a quattro dimensioni, il cosiddetto insieme dei quaternioni, indicato con H. Tra i molti usi per i quali vengono utilizzati faremo riferimento all'ambito delle rotazioni; si può dire che questi numeri descrivono in un modo intrinseco e naturale una rotazione nello spazio, con l'uso di matrici di rotazione.

Hamilton riuscì a trovare la struttura del suo insieme ragionando che partendo da " $i^2 = -1$ " devono esserci in questo insieme nuovi oggetti e introdusse altre due unità immaginarie, j e k. Questo spazio quindi è munito di tre unità immaginarie e l'equazione che lo descrive è " $i^2 = j^2 = k^2 = ijk = -1$ ".

## 4.2 ANGOLI DI EULERO

Gli angoli di Eulero descrivono la posizione di un sistema di riferimento XYZ solidale con un corpo rigido attraverso una serie di rotazioni a partire da un sistema di riferimento fisso x-y-z. I due sistemi di riferimento coincidono nell'origine.



Se i piani x-y e X-Y sono distinti, si intersecano in una retta (passante per l'origine) detta linea dei nodi (N). Se i piani coincidono, si definisce la linea dei nodi N come l'asse X.

Gli angoli di Eulero sono i seguenti:

- $\alpha$  è l'angolo tra l'asse x e la linea dei nodi. Detto angolo di precessione;
- $\beta$  è l'angolo tra gli assi z e Z. Detto angolo di nutazione;
- $\gamma$  è l'angolo tra la linea dei nodi e l'asse X. Detto angolo di rotazione propria;

Dal punto di vista dell'algebra lineare il passaggio dal sistema di riferimento fisso x-y-z a quello ruotato X-Y-Z equivale ad operare un cambiamento di base.

### **4.3 FUNZIONAMENTO ALGORITMO**

Abbiamo in INPUT i seguenti valori:

**accelerometerReading**- Letture dell'accelerometro nel sistema di coordinate del corpo del sensore ( $m/s^2$ ) matrice  $N \times 3$ . Le colonne della matrice corrispondono agli assi x -y - z del corpo del sensore. Le righe nella matrice  $N$  corrispondono a singoli campioni.

Tipo di dati: single | double

**magnetometerReading**- Letture del magnetometro nel sistema di coordinate del corpo del sensore ( $\mu T$ ) definiti da una matrice  $N \times 3$ . Le colonne della matrice corrispondono agli assi x -, y - e z del corpo del sensore. Le righe nella matrice  $N$  corrispondono ai singoli campioni.

Tipo di dati: single | double

Abbiamo in OUTPUT i seguenti valori:

**Orientation**: si passa da un sistema di coordinate globale a un sistema di coordinate del corpo, restituito come vettore di quaternioni o matrice.

Tipo di dati: quaternion | single | double

## **CONCLUSIONE**

Ecompass è una funzione di un algoritmo che restituisce una matrice di quaternione o rotazione e può ruotare le quantità da un frame principale (parent) verso un frame secondario (child). Per entrambi i formati di orientamento dell'output, l'operatore di rotazione viene determinato calcolando la matrice di rotazione.

$$R = \begin{bmatrix} (a \times m) \times a & a \times m & a \end{bmatrix}$$

Figura 8 Calcolo Matrice Rotazione

Dove “a” sono i dati appartenenti ad “accelerometerReading” e “m” ai dati appartenenti ad “magnetometerReading” entrambi in input. Per comprendere il calcolo della matrice di rotazione, si considera un punto arbitrario sulla Terra e il suo corrispondente frame NED (Nord Est Down) locale ipotizzando un sensore con struttura [x, y, z], avente la stessa origine del frame.

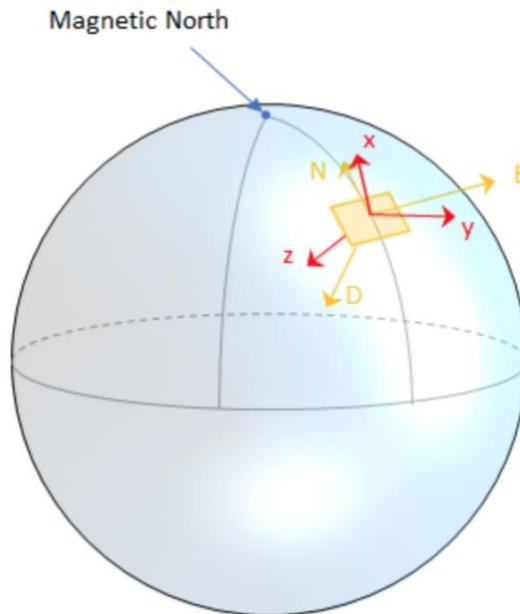


Figura 9

Ricordiamo che l'orientamento di un corpo della struttura del sensore è definito come l'operatore di rotazione (matrice di rotazione o quaternione) necessario per far ruotare una quantità da un frame Parent(NED) a un frame child (corpo del sensore):

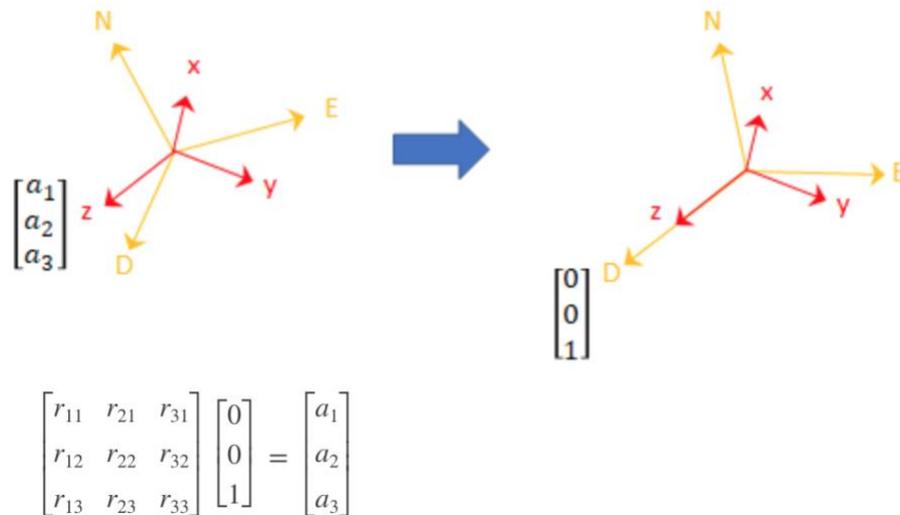
$$\begin{bmatrix} R \end{bmatrix} \begin{bmatrix} P_{\text{parent}} \end{bmatrix} = \begin{bmatrix} P_{\text{child}} \end{bmatrix}$$

Dove:

- R è una matrice di rotazione, che può essere interpretata come l'orientazione del frame child.
- P parent è un vettore 3x1 in nella struttura parent.
- P child è un vettore 3x1 nella struttura child.

Un accelerometro restituisce come risposta un'accelerazione dovuta alla gravità. Se la struttura del sensore è perfettamente

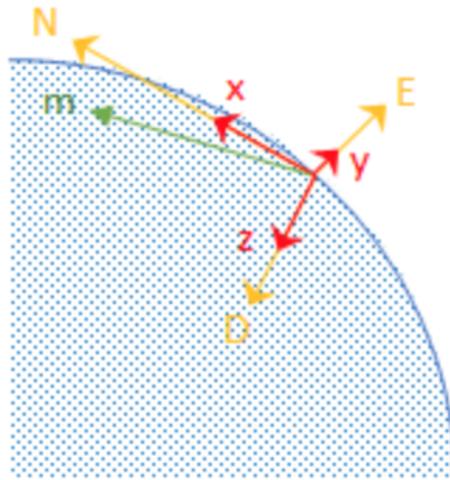
allineata con il sistema di coordinate NED, tutta l'accelerazione dovuta alla gravità va verso l'asse z e l'accelerometro indica  $[0 \ 0 \ 1]$ . Si considera la matrice di rotazione richiesta per far ruotare una quantità dal sistema di coordinate NED a una quantità indicata dall'accelerometro.



La terza colonna della matrice di rotazione corrisponde alle letture dell'accelerometro.

$$\begin{bmatrix} r_{31} \\ r_{32} \\ r_{33} \end{bmatrix} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

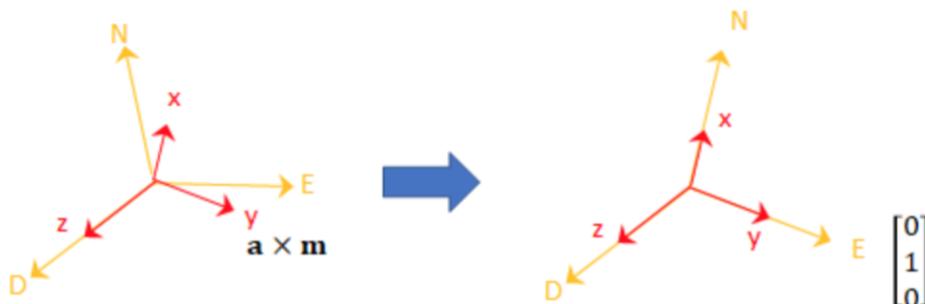
Le letture del magnetometro puntano verso il nord magnetico e si trova nel piano N – D. Anche il magnetometro sarà allineato alle coordinate del sistema NED.



Per definizione, l'asse E è perpendicolare al piano N - D. Se la struttura del sensore è allineata con il piano NED, sia il vettore accelerazione appartenente all'accelerometro e il vettore campo magnetico appartenente al magnetometro restano nel piano N - D.

Pertanto “ $m \times a = y$ ”, sempre con un certo ridimensionamento dell'ampiezza.

- Si considera la matrice di rotazione richiesta per ruotare NED sul frame figlio, [x y z].



$$\begin{bmatrix} r_{11} & r_{21} & r_{31} \\ r_{12} & r_{22} & r_{32} \\ r_{13} & r_{23} & r_{33} \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \times \begin{bmatrix} m_1 \\ m_2 \\ m_3 \end{bmatrix}$$

La seconda colonna della matrice di rotazione corrisponde al prodotto incrociato tra “accelerometer reading” e “magnetometer reading”:

$$\begin{bmatrix} r_{21} \\ r_{22} \\ r_{23} \end{bmatrix} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \times \begin{bmatrix} m_1 \\ m_2 \\ m_3 \end{bmatrix}$$

**Figura 10** a(accelerometer), m(magnetometer)

Ed infine per definizione, la colonna 1 della matrice di rotazione è il prodotto tra la colonna 2 e la colonna 3:

$$\begin{aligned} \begin{bmatrix} r_{11} \\ r_{12} \\ r_{13} \end{bmatrix} &= \begin{bmatrix} r_{21} \\ r_{22} \\ r_{23} \end{bmatrix} \times \begin{bmatrix} r_{31} \\ r_{32} \\ r_{33} \end{bmatrix} \\ &= (a \times m) \times a \end{aligned}$$

**Figura 11:** Per ottenere la colonna 1

Si evince che l’algoritmo è suscettibile al rumore del sensore.

[4]

## **CAPITOLO 5: DATA FUSION**

Una delle domande che solitamente ci si pone prima di fare qualsiasi cosa è banalmente: “Perché?”. Come disse Lord William Thomson «possiamo conoscere qualcosa dell’oggetto di cui stiamo parlando solo se possiamo eseguirvi misurazioni, per descriverlo mediante numeri; altrimenti la nostra conoscenza è scarsa e insoddisfacente».

[1]

### **5.1 COSA SIGNIFICA**

Significa combinare in modo intelligente i dati provenienti da diversi sensori e diversi algoritmi in modo da ottenere informazioni migliori di quelle che si sarebbero potute ottenere usando quei sensori separatamente. [2]

### **5.2 IL MIO ALGORITMO DI DATA FUSION**

Ho creato un algoritmo che consente di stimare l’orientazione attraverso 3 algoritmi di data fusion, in modo da vedere le principali differenze che il rumore dei sensori può creare all’interno dell’analisi del segnale.

Il confronto sarà effettuato con i dati precedentemente illustrati di “rotazione sx – dx- 90°” e “acquisizione 30 minuti”. Per tutti i sensor

fusion vale una frequenza di campionamento (FS) di 20Hz e fattore di decimazione uguale a 2 per ridurre il costo computazionale.

Il codice del mio algoritmo è il seguente:

```
%accelerometerReading = importdata("accelerometro.txt" );
%magnetometerReading = importdata("magnetometro.txt");
orientation = ecompass(accelerometerReading,magnetometerReading);
rotationvector=rotvecd (orientation);

quaternionEulerAngles = eulerd (orientation, 'ZYX' , 'frame' )

gravityVectors = rotatepoint (orientation,accelerometerReading );

plot (gravityVectors);
timeVector = (0: 35994-1) / Fs;
figure(1)
plot(timeVector,quaternionEulerAngles);
title ('Angoli Eulero')
legend('z-axis','y-axis','x-axis')
xlabel('Time (s)')
ylabel('Rotation (degrees)')

decim = 2;
fuse = imufilter('SampleRate',Fs,'DecimationFactor',decim);
q = fuse(accelerometerReading,gyroscopeReading);
time = (0:decim:size(accelerometerReading,1)-1)/Fs;
figure(2)
plot(time,eulerd(q,'ZYX','frame'))
title('Orientation Estimate with IMU FILTER')
legend('Z-axis', 'Y-axis', 'X-axis')
xlabel('Time (s)')
ylabel('Rotation (degrees)')

decim1 = 2;
fuse1 = ahrsfilter('SampleRate',Fs,'DecimationFactor',decim);
q1 = fuse1(accelerometerReading,gyroscopeReading,magnetometerReading);
time1 = (0:decim:size(accelerometerReading,1)-1)/Fs;
figure(3)
plot(time1,eulerd(q1,'ZYX','frame'))
title('Orientation Estimate with AHRSFILTER ')
legend('z-axis', 'y-axis', 'x-axis')
ylabel('Rotation (degrees)')
```

Si noti che al suo interno, oltre all'algoritmo Ecompass, sono presenti due algoritmi di data fusion ai quali è applicato un filtro di Kalman:

- Imufilter
- Ahrsfilter

È stato scelto di applicare un filtro di Kalman perché ci si è accorti che il rumore generato dall'algoritmo e-compass non consente una così precisa analisi e stima della posizione nel tempo.

### **IMUFILTER**

“Imufilter” fonde le letture dell'accelerometro e del giroscopio per stimare l'orientazione del corpo e applica un filtro di Kalman sulla fusione dei due dati.

Abbiamo in INPUT:

**accelerometerReading**- Letture dell'accelerometro nel sistema di coordinate del corpo del sensore ( $m/s^2$ ) definiti da una matrice  $N \times 3$ . Le colonne della matrice corrispondono agli assi x-y-z dell'accelerometro. Le righe nella matrice  $N$  corrispondono a singoli campioni.

**gyroscopeReading**- Letture del giroscopio nel sistema di coordinate del corpo del sensore in (rad/s) definiti da una matrice  $N \times 3$ . Le colonne della matrice corrispondono agli assi x-y-z

dell'accelerometro. Le righe nella matrice N corrispondono ai singoli campioni.

Il tutto alla frequenza di campionamento (FS) precedentemente descritta.

Abbiamo in OUTPUT:

**Orientation**- che può ruotare le quantità da un sistema di coordinate globale a un sistema di coordinate rispetto al corpo, il tutto restituito come quaternioni. Si è trasformato l'output in angoli di Eulero per semplicità di calcolo tramite la funzione "eulerd".

Il data fusion stima correttamente il cambiamento di orientamento rispetto a un presunto orientamento iniziale rivolto a nord. Tuttavia, l'asse x del dispositivo puntava verso sud quando registrato. Per avere un miglior orientamento rispetto a quello iniziale si usi "ahrsfilter". [5]

#### **PROPRIETÀ FILTRO:**

- ACCELLEROMETRO:

Varianza del rumore del segnale =  $0.00019247 (m / s^2)^2$

- MAGNETOMETRO:

Varianza del rumore del segnale =  $0.1 \mu T^2$

- GIROSCOPIO:

Varianza del rumore del segnale =  $9.1385e-5 (rad / s)^2$

## **AHRSFILTER**

“Ahrsfilter” fonde le letture dell’accelerometro, del magnetometro e del giroscopio per stimare l’orientazione del corpo e applica un filtro di Kalman sulla fusione dei tre dati.

Abbiamo in INPUT:

**accelerometerReading**- Letture dell'accelerometro nel sistema di coordinate del corpo del sensore ( $m/s^2$ ) definiti da una matrice  $N \times 3$ . Le colonne della matrice corrispondono agli assi  $x$ - $y$ - $z$  dell’accelerometro. Le righe nella matrice  $N$  corrispondono a singoli campioni. Il tutto alla frequenza di campionamento (FS) precedentemente descritta.

**gyroscopeReading**- Letture del giroscopio nel sistema di coordinate del corpo del sensore in ( $rad/s$ ) definiti da una matrice  $N \times 3$ . Le colonne della matrice corrispondono agli assi  $x$ - $y$  e  $z$  dell’accelerometro. Le righe nella matrice  $N$  corrispondono ai singoli campioni.

**magnetometerReading**- Letture del magnetometro nel sistema di coordinate del corpo del sensore ( $\mu T$ ) definiti da una matrice  $N \times 3$ . Le colonne della matrice corrispondono agli assi  $x$ -,  $y$ - e  $z$  del corpo del sensore. Le righe nella matrice  $N$  corrispondono ai singoli campioni.

Il tutto alla frequenza di campionamento (FS) precedentemente descritta.

Abbiamo in OUTPUT:

**Orientation**- che può ruotare le quantità da un sistema di coordinate globale a un sistema di coordinate rispetto al corpo, il tutto restituito come quaternioni. Si è trasformato l'output in angoli di Eulero per semplicità di calcolo tramite la funzione "eulerd".

Il fusion stima correttamente la variazione dell'orientamento nel tempo, incluso l'orientamento iniziale rivolto a sud. Tra i vari filtri e algoritmi studiati è quello che ci da risultati meno rumorosi e precisi.  
[6]

PROPRIETÀ FILTRO:

- ACCELEROMETRO:

Varianza del rumore del segnale =  $0.00019247 \text{ (m / s }^2)^2$

- MAGNETOMETRO:

Varianza del rumore del segnale =  $0.1 \mu\text{T}^2$

- GIROSCOPIO:

Varianza del rumore del segnale =  $9.1385\text{e-}5 \text{ (rad / s)}^2$

## 5.3 RISULTATI

### Algoritmo data fusion applicato ad “Acquisizione Destra e sinistra”:

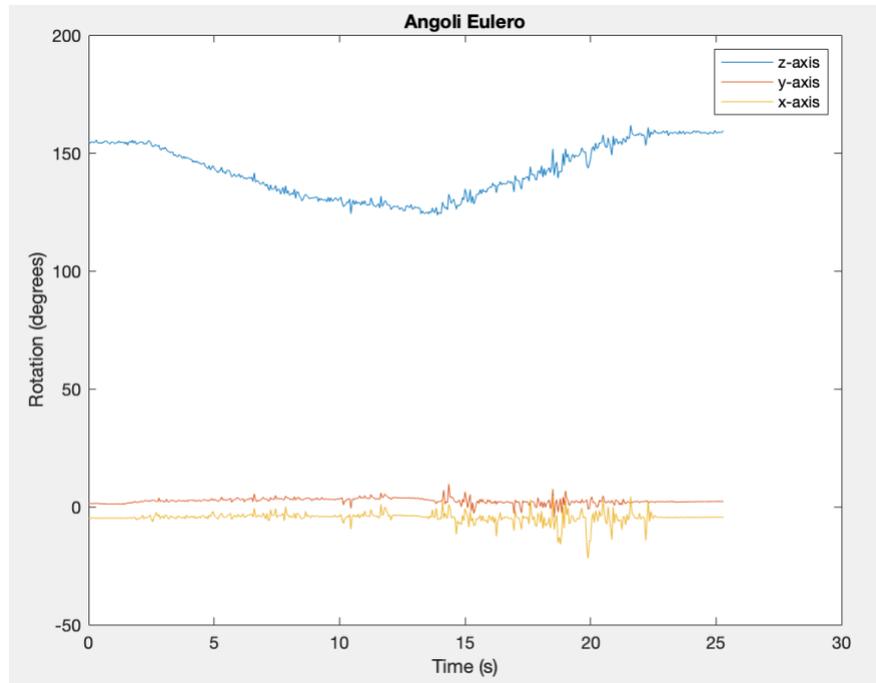


Figura 12: Output Algoritmo Ecompass

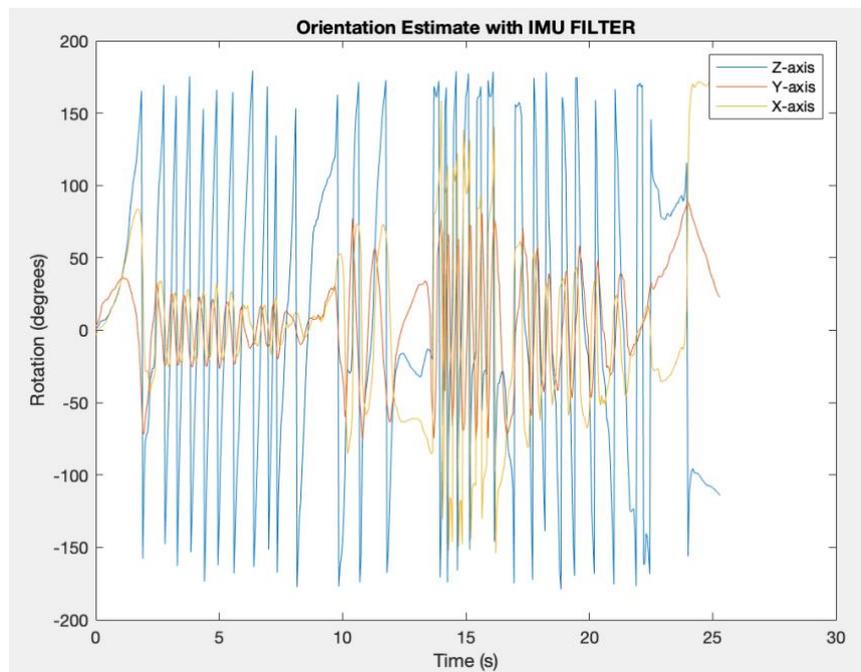
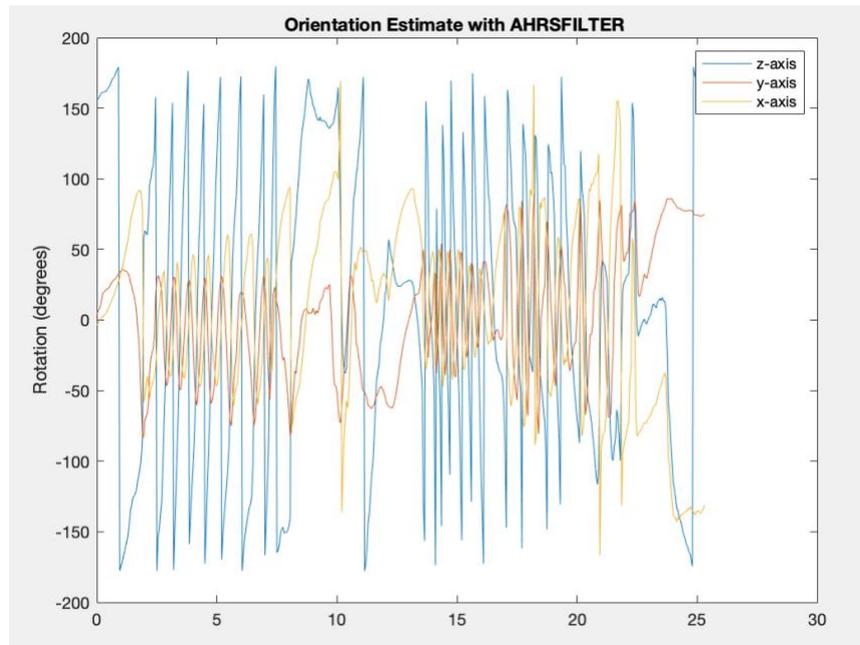


Figura 13: Output algoritmo imufilter



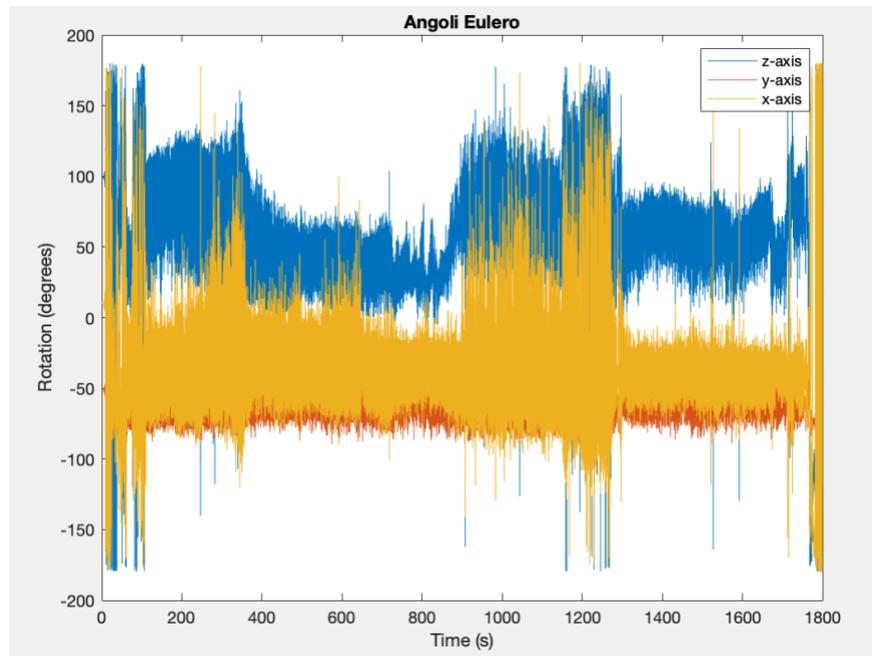
**Figura 14: Output algoritmo Ahrsfilter**

Si evince facendo un confronto con la [figura 4](#) (Output Acquisizione sinistra destra e ritorno) che siamo riusciti a far diminuire il rumore presente nei segnali dei sensori in modo da avere un'affidabile stima del corpo nel tempo.

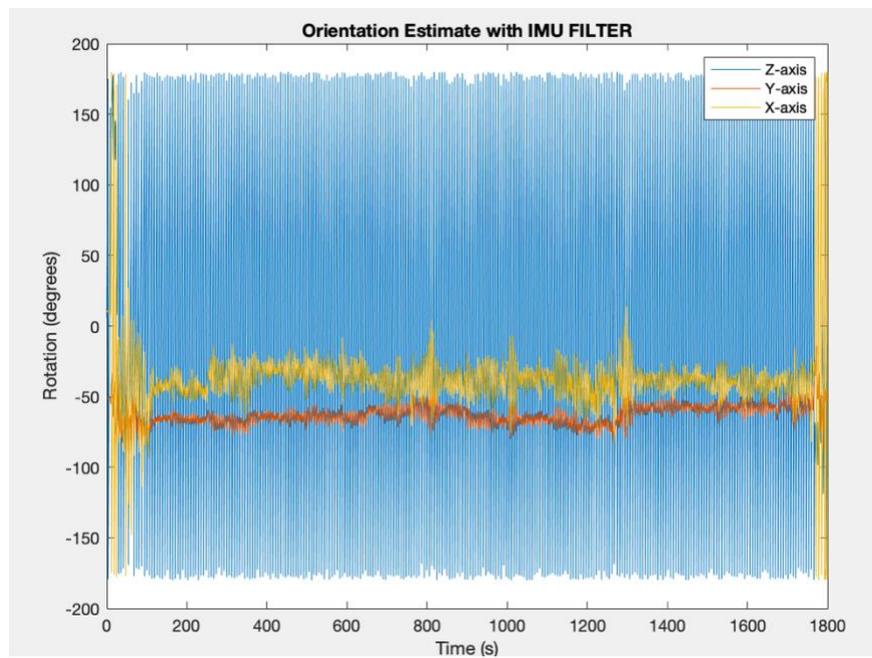
Gli algoritmi si dimostrano efficienti per un tipo di acquisizione di movimenti semplici e di breve durata.

La situazione potrebbe complicarsi analizzando acquisizioni con durata maggiore e movimenti non lineari.

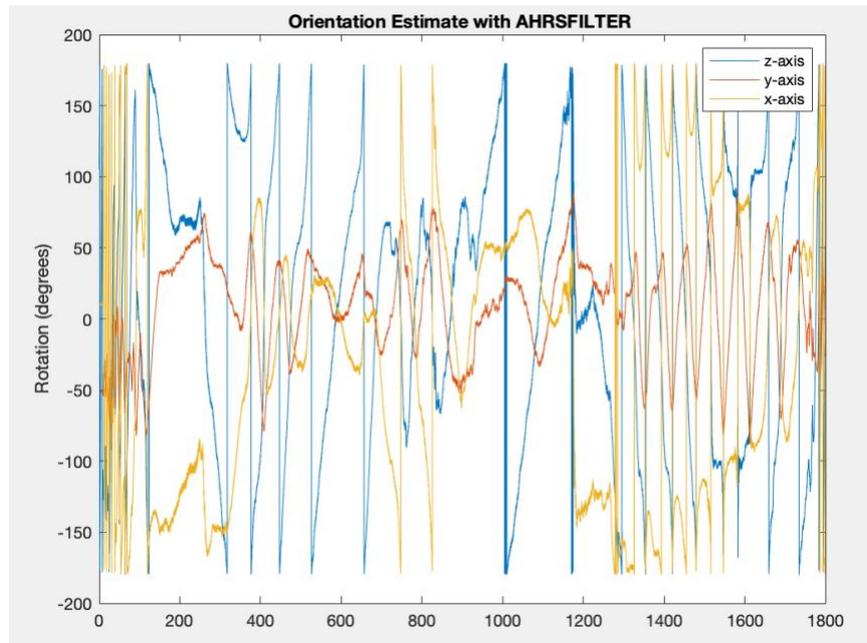
## Algoritmo data fusion applicato ad “Acquisizione 30 minuti”:



**Figura 15: Output algoritmo Ecompass**



**Figura 16: Output Algoritmo Imufilter**



**Figura 17: Output algoritmo ahrsfilter**

Si evince facendo un confronto con la [figura 7](#) (Acquisizione 30 minuti) che aumentando la durata e la molteplicità dei segnali, gli algoritmi risultano comunque affidabili nella previsione della posizione nel corpo nel tempo.

Gli algoritmi e-compass e imufilter risultano particolarmente influenzati dal rumore del giroscopio, mentre nell'algoritmo ahrsfilter questo problema va a diminuire.

Una possibile soluzione futura potrebbe essere quella di applicare un filtro complementare (passa basso) al segnale del giroscopio.

## **CONCLUSIONI:**

Nella tesi è stata introdotta un'analisi di segnali presi da sensori IMU. Si è cercato di trattarli nel modo migliore possibile cercando di stimare nel modo più corretto e preciso l'orientazione del corpo nel tempo e nello spazio tenendo conto del rumore e dei tipi di segnale, sapendo che ogni sensore reagisce a modo proprio in base alla durata del segnale analizzato.

Si è avuto a disposizione un ampio database di acquisizione di dati in tempo reale descritti da una dozzina di tipi di diversi movimenti. Di questi non era possibile effettuare nessuna stima. Sono stati scelti per lo studio approfondito dell'algoritmo di data fusion i movimenti che secondo vari confronti potevano far rendere di più l'idea di quello che si stava facendo, confrontando acquisizioni semplici e acquisizioni complesse con lo scopo di far emergere i principali problemi.

I risultati in output delle tre diverse analisi svolte confermano la loro compatibilità con il segnale delle acquisizioni in tempo reale. Gli algoritmi di data fusion sono molto efficaci perché sono molto robusti rispetto alle fonti di rumore ambientale. Il risultato più affidabile e preciso per la stima dell'orientazione ci è dato dall'algoritmo "ahrsfilter" che riduce notevolmente l'errore verso i poli N-E-D. Con l'implementazione degli algoritmi "imufilter" e

“ahrsfilter” è stato studiato anche il funzionamento del filtro di Kalman e del suo comportamento su segnali di determinati sensori. Una possibile implementazione sarebbe quella di applicare un filtro passa basso ai segnali acquisiti del giroscopio, dato che è il sensore più influenzato dal rumore e quello dove il filtraggio di Kalman è meno efficace, per poi quindi creare un altro algoritmo di data fusion per la stima dell’orientamento.

## **BIBLIOGRAFIA**

[1] "Grandezze fisiche e loro misura" Materiale didattico Corsi Zero, Università degli studi di Enna "Kore":

<http://www.unikore.it/index.php/ingegneria-informatica-attività-didattiche/materiale-didattico-corsi-zero/>

[2] University of Bonn, Institute of Computer Science 4: [net.cs.uni-bonn.de/wg/sensor-data-and-information-fusion/what-is-it/](http://net.cs.uni-bonn.de/wg/sensor-data-and-information-fusion/what-is-it/)

[10] [https://dipmat.univpm.it/~demeio/public/Meccanica%20Razionale%20-%20Materiale/Angoli\\_Eulero\\_Wiki\\_1.pdf](https://dipmat.univpm.it/~demeio/public/Meccanica%20Razionale%20-%20Materiale/Angoli_Eulero_Wiki_1.pdf)

## **SITOGRAFIA**

[3] <https://it.mathworks.com/help/fusion/ref/ecompass.html>

[4] <https://it.mathworks.com/help/fusion/ref/imufilter-system-object.html>

[5] <https://it.mathworks.com/help/fusion/ref/ahrsfilter-system-object.html>

[6] <https://it.mathworks.com/help/fusion/ref/ecompass.html>

[7] <https://www.settorezero.com/wordpress/cosa-sono-come-funzionano-e-a-cosa-servono-gli-accelerometri/>

[8] <https://www.militarypedia.it/come-funziona-il-giroscopio/>

[9] <https://www.thedifferentgroup.com/2017/02/10/giroscopio-come-funziona/>