



UNIVERSITÀ
POLITECNICA
DELLE MARCHE

Corso di Laurea in Ingegneria Informatica e dell'Automazione

**SVILUPPO DI ALGORITMI DI DATA FUSION
PER PIATTAFORME INERZIALI BASATI SU
FILTRI COMPLEMENTARI**

DEVELOPMENT OF DATA FUSION ALGORITHMS FOR INERTIAL PLATFORMS BASED
ON COMPLEMENTARY FILTERS

Relatore: Chiar.mo

PAOLO CRIPPA

Tesi di Laurea di:

FABIO SILVERI

Correlatore: Chiar.mo

GIORGIO BIAGETTI

A.A. 2018/2019

INDICE

Capitolo 1: INTRODUZIONE

1.1 Sensor Fusion

Capitolo 2: MATLAB

Capitolo 3: PIATTAFORME IMU

3.1 Cosa sono

3.2 Accelerometro

3.3 Giroscopio

3.4 Magnetometro

3.5 Sensore utilizzato

Capitolo 4: PITCH, ROLL, YAW

Capitolo 5: ALGORITMI E DATI DI PARTENZA

5.1 Raccolta dati

5.2 Raccolta e analisi dati sulla traslazione verso destra e ritorno

5.3 Raccolta e analisi dei dati sull'acquisizione di 30 minuti

5.4 Problematiche

Capitolo 6: FILTRI

6.1 Filtri principali

6.2 Filtro complementare

Capitolo 7: DATA FUSION

7.1 Data fusion traslazione destra e ritorno

7.2 Data fusion acquisizione 30 minuti

Capitolo 8: CONCLUSIONI

RIFERIMENTI

INDICE DELLE EQUAZIONI

INDICE DELLE FIGURE

1. INTRODUZIONE

1.1 *Data Fusion*

La fusione dei dati dei sensori è un fenomeno onnipresente che esisteva prima della sua realizzazione tecnologica. Tutte le creature viventi, compresi gli esseri umani, per natura eseguono la fusione dei dati dei sensi. Ciascuno a suo modo, combina o "fonde" le sensazioni fornite da differenti organi di senso con conoscenze apprese da esperienze precedenti e interazioni con altre creature. Di conseguenza, producono un "quadro mentale" del loro ambiente individuale, la base per comportarsi in modo appropriato nella loro lotta per evitare danni o raggiungere con successo un determinato obiettivo in una determinata situazione.

Il moderno data fusion mira ad automatizzare questa capacità di combinare pezzi complementari di informazione. La fusione dei dati del sensore produce quindi un "quadro della situazione", una ricostruzione di una "situazione reale" sottostante, resa possibile da un'attuazione efficiente algoritmi matematici che sfruttano anche i dati imperfetti e arricchiti da nuove fonti di informazione.

I risultati mostrano che in poche righe di codice è possibile ottenere un algoritmo di data fusion basato su filtri complementari e capace di calcolare in modo efficiente l'inclinazione dello stesso sensore IMU.

2. MATLAB



Il software utilizzato per l'elaborazione dei dati e lo sviluppo dell'algoritmo di data fusion è *MATLAB* (abbreviazione di *Matrix Laboratory*) (Figura 1). *Matlab* è un ambiente per il calcolo numerico e l'analisi numerica scritto in C, comprendente il linguaggio di programmazione ideato dalla *MathWorks*. *MATLAB* fu creato alla fine degli anni Settanta da Cleve Moler, presidente del dipartimento di scienze informatiche dell'Università del Nuovo Messico. L'ingegnere Jack Little conobbe il programma durante una visita a Mole, e, riconoscendo il suo potenziale commerciale, si unì con Moler. I due riscrissero *MATLAB* in linguaggio C e fondarono la *MathWorks* nel 1984 per continuare il suo sviluppo.

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 (Mathworks, 2020).

3. PIATTAFORMA IMU

3.1 *Che cos'è la piattaforma IMU*

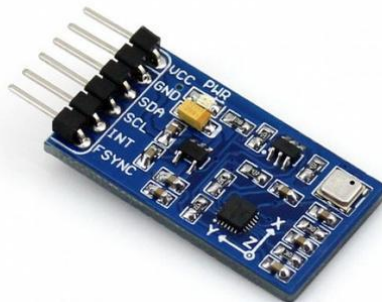


Figura 2: piattaforma IMU

L'IMU, o Unità di Misura Inerziale, è un dispositivo elettronico (Figura 2) che misura accelerazioni, velocità di rotazione e possibilmente campo magnetico terrestre con l'uso dell'accelerometro triassiale, del giroscopio triassiale e talvolta del magnetometro triassiale al fine di determinare l'orientamento di un oggetto. Grazie alle loro caratteristiche uniche, i dispositivi IMU sono stati a lungo oggetto di approfondimento di ricerca nei settori aerospaziali e di navigazione, sebbene inizialmente le dimensioni erano ingombranti. Negli ultimi anni, con l'avvento di IMU basati su MEMS (Micro-Electro-Mechanical-system), la dimensione dei sensori IMU si è drasticamente ridotta alla dimensione del chip. Quest'ultimo, insieme alla riduzione dei costi e del consumo di energia, ha

notevolmente accelerato molte attività di ricerca & sviluppo e di lavoro. Robotica e analisi del movimento umano sono le aree maggiormente beneficiate.

I sensori magneto inerziali sono sistemi elettronici composti da:

- un giroscopio triassiale
- un accelerometro triassiale.

Spesso si aggiunge alla piattaforma IMU un magnetometro triassiale: tale configurazione prende il nome di “MARG” (Magnetic Angular Rate and Gravity). Le informazioni sull'orientamento del sensore sono ottenute dall'integrazione delle misure di velocità angolari e descritte mediante gli angoli di Eulero. Le misure del giroscopio sono però affette da un errore di *bias* che comporta un progressivo deterioramento delle stime degli angoli calcolati. Una possibile correzione dei dati raccolti può essere attuata sfruttando le stime di accelerazione fornite dall'accelerometro e di campo magnetico dal magnetometro mediante appositi algoritmi Sensor Fusion. Questo sinergico utilizzo di tre tipi diversi di sensori nella stessa piattaforma rende molto versatile la configurazione MARG.

I dati acquisiti dalle piattaforme IMU devono essere elaborati (filtrati) per produrre le informazioni di assetto. Ci sono diversi metodi e tecniche impiegate. Ne sono un esempio le tecniche di filtraggio che si basano su algoritmi di sensor fusion basati su filtri complementari, o con il filtro di Kalman.

3.2 Accelerometro

L'accelerometro (Figura 3) è uno 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). 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 un'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 moderni sistemi di misura.

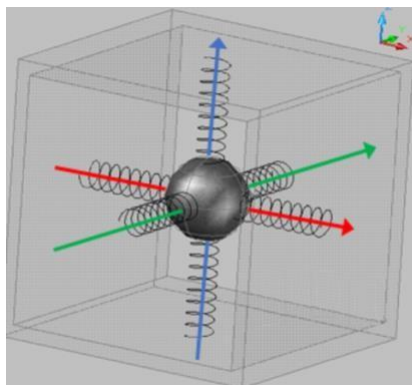


Figura 3: esempio accelerometro

3.3 Giroscopio

Un giroscopio misura la velocità di rotazione sugli assi X, Y, Z calcolate tramite masse microscopiche, la cui 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, tanto da sembrare capace di sfidare la gravità. 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.

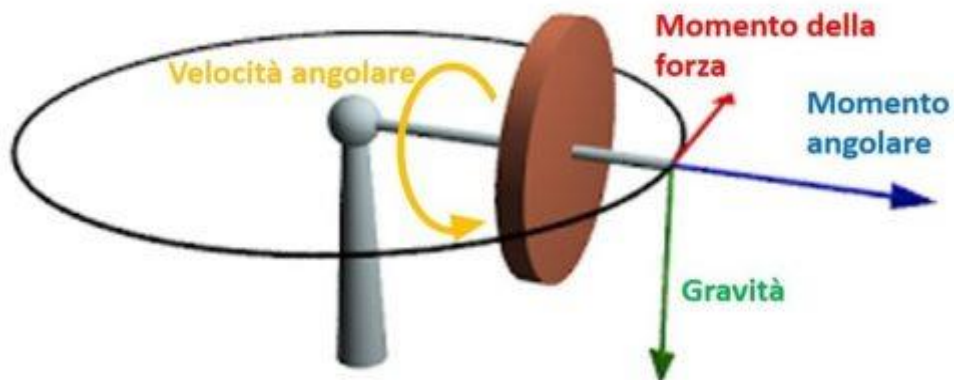


Figura 4: funzionamento giroscopio (Pinto, 2017)

Nella [Figura 4](#) 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 (τ). Quando il disco comincia a ruotare in senso antiorario, come indicato dalla freccia gialla della velocità angolare (ω) si crea un momento angolare (L) che può essere misurato secondo la formula seguente ([Equazione 1](#)) :

$$L = I \times \omega$$

Equazione 1: momento angolare

Il momento angolare è il prodotto tra la velocità angolare (ω) e il momento di inerzia (I). L misura quindi l'inerzia del corpo al mutare della sua velocità angolare.

Secondo quanto afferma la legge di conservazione del momento angolare, il valore del momento angolare rimane costante se non viene applicata alcuna forza.

Matematicamente parlando questo è molto semplice da dimostrare. Il momento della forza (τ) ([Equazione 2](#)) che agisce su un sistema è pari alla velocità con cui il momento angolare (L) cambia nel tempo (t). Usando i simboli della matematica questo si traduce in:

$$\tau = dL/dt$$

Equazione 2: momento della forza

Se non c'è alcuna forza che agisce sul sistema avremo $\tau = 0$ ([Equazione 3](#)) ossia:

$$dL/dt = 0$$

Equazione 3: $\tau = 0$

Integrando questa [Equazione 3](#), L risulterà costante.

Cosa succede quindi ad un sistema quando viene impartito un movimento circolare? In questo caso l'effetto che si otterrà sarà che l'asse intorno a cui il sistema ruota cercherà di puntare sempre nella stessa direzione della rotazione. Riguardando la [Figura 4](#) possiamo immaginare la freccia rossa del momento della forza possa trainare verso di sé la freccia blu del momento angolare, facendo in modo che la ruota giri lungo la traiettoria circolare indicata nel disegno e vincendo la forza di gravità (e quindi il peso stesso dell'oggetto) che tenderebbe a far cadere il disco verso il basso (come mostrato dalla direzione della freccia verde) (Pinto, 2017).

3.4 Magnetometro

Il magnetometro misura 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 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. (Casolino, 2016)

3.5 Sensore utilizzato

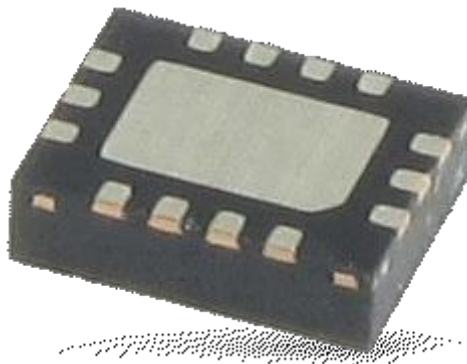


Figura 5: sensore LSM6DSM

Il sensore utilizzato è il LSM6DSM (Figura 5), prodotto dall'azienda STMicroelectronics, fornisce i dati come valori digitali a 16 bit.

I versi degli assi per le accelerazioni lineari e i sensi di rotazione degli stessi assi per le velocità angolari sono illustrati nella [Figura 6](#) e nella [Figura 7](#) prese dal datasheet del sensore.

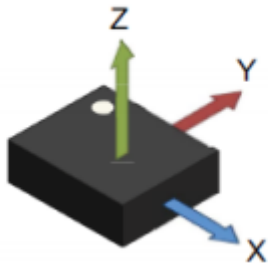


Figura 6: versi degli assi per le accelerazioni lineari

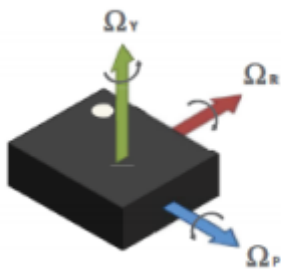


Figura 7: sensi di rotazione degli assi per le velocità angolari

4. PITCH, ROLL, YAW

Nel seguente capitolo saranno introdotti dei concetti relativi alla rotazione di un corpo nello spazio, per capire fino a che punto può essere utile un sensore IMU per la misurazione dell'inclinazione. Un corpo può rotare nello spazio lungo tre assi. Facendo riferimento alla dinamica applicata ai velivoli si parla più propriamente di rollio, beccheggio e imbardata.

Per convenzione, la rotazione del velivolo lungo l'asse X (in rosso) del velivolo si chiama rollio (Roll). La rotazione lungo l'asse Y del velivolo (in viola) si chiama beccheggio (Pitch) e la rotazione lungo l'asse Z del velivolo (in giallo) si chiama Imbardata (Yaw) (Figura 8).

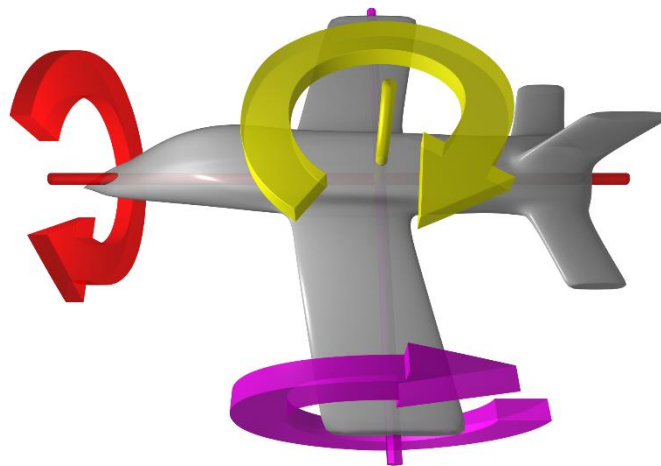


Figura 8: angoli di rollio (colore rosso), beccheggio (viola) e imbardata (giallo)

Immaginiamo ora che gli assi X e Y del velivolo si trovino sul piano dell'orizzonte e l'asse Z sia perpendicolare a questo piano.

Una rotazione intorno all'asse X forma un angolo tra asse Y e orizzonte. Questo angolo che Y forma con l'orizzonte a causa della rotazione di X si chiama angolo di rollio e viene generalmente indicato con la lettera greca ϕ .

Essendo l'asse Z ortogonale con l'asse Y, una rotazione intorno a X genera lo stesso angolo tra Z e il vettore della forza di gravità

Una rotazione intorno all'asse Y forma un angolo tra asse X e orizzonte. Questo angolo che X forma con l'orizzonte a causa della rotazione di Y si chiama angolo di beccheggio e viene generalmente indicato con la lettera greca θ .

La rotazione intorno all'asse Z non porta alla formazione di nessun nuovo angolo con l'orizzonte: la rotazione intorno a Z porta piuttosto gli assi X e Y a ruotare di un certo angolo sul piano in cui già si trovano. Questo angolo si chiama angolo di imbardata e viene generalmente indicato con la lettera greca ψ .

Un accelerometro a tre assi è in grado di rilevare l'angolo di rollio e l'angolo di beccheggio ma non l'angolo di imbardata per quanto appena detto. Infatti, la rotazione intorno a Z non provoca una variazione degli angoli che gli assi formano rispetto all'orizzonte e di conseguenza i valori di accelerazione gravitazionale a cui i singoli assi sono sottoposti.

Per tale motivo non è possibile usare un accelerometro nel caso in cui bisogna anche rilevare l'angolo di imbardata. Per fare questo si integra il giroscopio.

I giroscopi sono indispensabili sugli elicotteri per evitare la rotazione indesiderata intorno all'asse di rotazione dell'elica principale. Un altro sistema, meno preciso, per rilevare l'angolo di imbardata potrebbe essere quello di utilizzare un magnetometro. (Bernardo, 2011)

5. ALGORITMI E DATI DI PARTENZA

5.1 Raccolta dati

La piattaforma inerziale IMU ha registrato una serie di movimenti:

- Acquisizione 30 minuti;
- Aumento e diminuzione quota;
- Inclinazione avanti – indietro e ritorno;
- Inclinazione sinistra – destra e ritorno;
- Rotazione di 360°;
- Rotazione sinistra di 90° e ritorno;
- Traslazione verso destra e ritorno.

Per ogni serie di movimenti sono stati creati tre datasheet, uno che raccoglie i dati dell'accelerometro ([Figura 9](#)), uno che raccoglie i dati del giroscopio ([Figura 10](#)) e uno che raccoglie i dati del magnetometro ([Figura 11](#)). Ogni datasheet ha la medesima composizione:

- Prima colonna: timeStamp di ogni acquisizione;
- Seconda colonna: Orario di ogni acquisizione;
- Terza colonna: Inclinazione sull'asse delle X;
- Quarta colonna: Inclinazione sull'asse delle Y;
- Quinta colonna: Inclinazione sull'asse delle Z.

21567	084327.977706	0.775390625	0.1123046875	0.599609375
21573	084328.023409	0.771484375	0.1044921875	0.615234375
21579	084328.082595	0.7783203125	0.1103515625	0.6064453125
21586	084328.127728	0.771484375	0.11328125	0.6064453125
21592	084328.173422	0.7763671875	0.1044921875	0.6162109375
21598	084328.232617	0.775390625	0.1162109375	0.5986328125
21604	084328.277708	0.7734375	0.1083984375	0.619140625

Figura 9: raccolta dati accelerometro

21567	084327.977706	-0.6	1.5	0.7
21573	084328.023409	0.9	0.4	0.3
21579	084328.082595	-0.2	1.4	0.3
21586	084328.127728	1.3	0.2	0.4
21592	084328.173422	-0.8	1.9	0.4
21598	084328.232617	1.6	-0.1	0.2
21604	084328.277708	-0.1	1.4	0.5

Figura 10: raccolta dati giroscopio

21567	084327.977706	-10.986328125	-307.6171875	118.10302734375
21573	084328.023409	-9.46044921875	-307.6171875	115.3564453125
21579	084328.082595	-6.40869140625	-308.53271484375	116.5771484375
21586	084328.127728	-9.46044921875	-310.6689453125	119.93408203125
21592	084328.173422	-7.62939453125	-307.6171875	117.49267578125
21598	084328.232617	-7.62939453125	-310.36376953125	118.408203125
21604	084328.277708	-6.7138671875	-308.837890625	118.408203125

Figura 11: raccolta dati magnetometro

5.2 Raccolta e analisi dati sulla traslazione verso destra e ritorno

Al fine di acquisire i dati necessari per quanto riguarda la traslazione verso destra e ritorno, questi sono stati registrati tramite una traslazione del sensore IMU verso destra e successivo ritorno al punto di partenza. Il sensore è stato posizionato su un piano orizzontale.

L'algoritmo per restituire in output i dati raccolti è il seguente (Figura 12):

```

clc
close all
clear all
format long
acc_matrix = importdata ("Accelerometro (traslazione 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 (traslazione 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')

```

```
magn_matrix = importdata ("Magnetometro (traslazione 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 [µT]')
title('Magnetometro')

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);
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 [µT]')
```

Figura 12: algoritmo di partenza traslazione verso destra e ritorno

L'algoritmo esegue il seguente output (Figura 13):

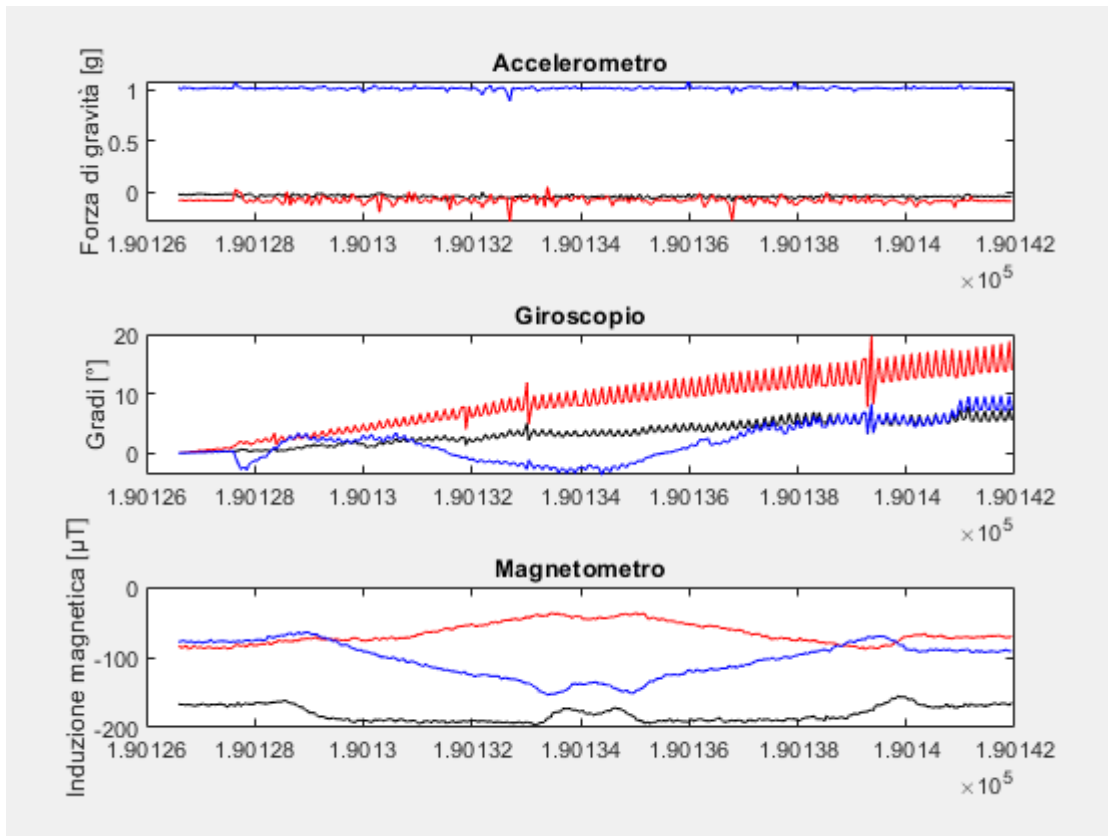


Figura 13: output algoritmo traslazione verso destra e ritorno

5.3 Raccolta e analisi dei dati sull'acquisizione di 30 minuti

Al fine di acquisire i dati necessari per quanto riguarda l'acquisizione del percorso di 30 minuti, questi sono stati registrati effettuando un percorso

a piedi di 30 minuti attorno al complesso universitario di Monte Dago (Ancona) (Figura 14).



Figura 14: percorso e rappresentazione grafica dell'andamento

Il sensore è stato posizionato all'interno di uno zaino, trasportato con il passo più omogeneo possibile. È da tenere a mente che il percorso è composto da dislivelli e da andamenti non uniformi, aggravanti che ne hanno influenzato l'analisi del segnale tramite l'algoritmo della Figura 15.

```
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)];
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
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 gravità [g]')
legend ('Asse x', 'Asse y', 'Asse z')
title('Accelerometro')

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;
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
        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 [°]')
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 [uT]')
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 gravità [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 [°]')

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 [µT]')

```

Figura 15: algoritmo acquisizione 30 minuti

L'algoritmo esegue il seguente output (Figura 16):

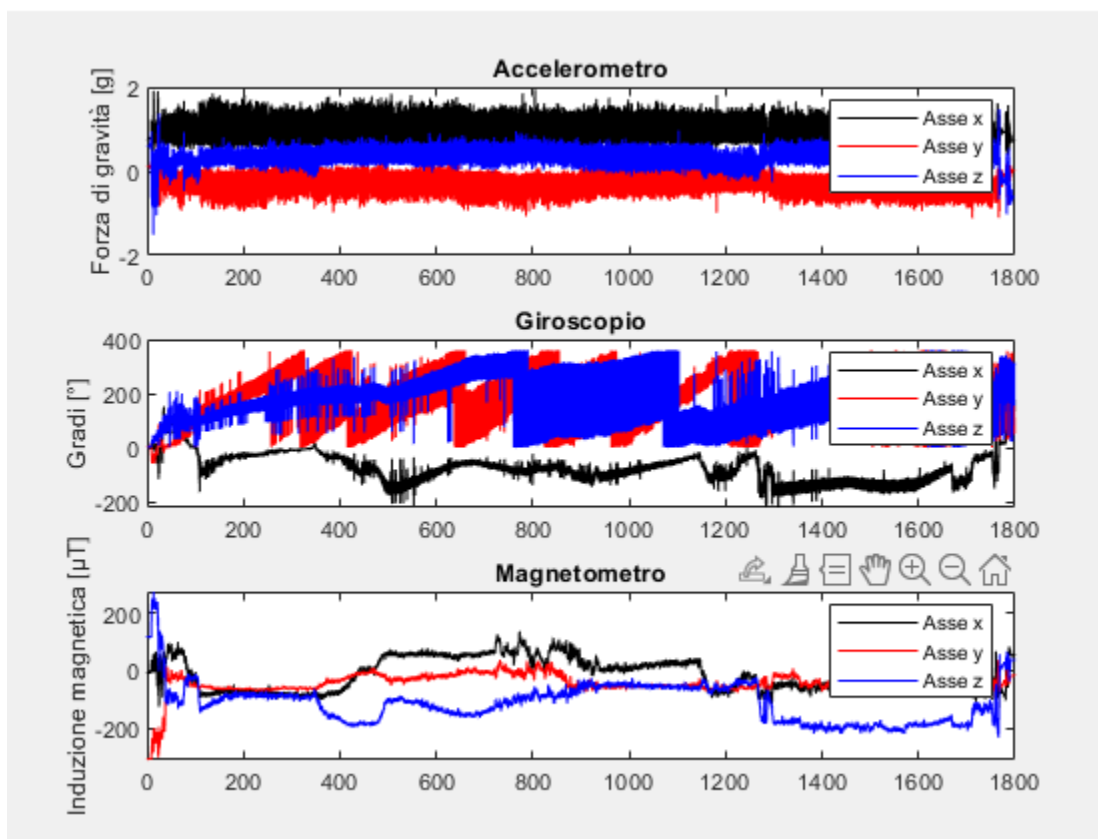


Figura 16: output algoritmo acquisizione 30 minuti

5.4 Problematiche relative ai dati acquisiti

Analizzando i dati si è rilevato un problema dovuto alla variazione del bias del giroscopio. Gli output del giroscopio sono soggetti a variazioni di offset quando sottoposti ad accelerazioni lineari, che a causa dell'integrazione si trasformano in fenomeni di drift.

Durante la camminata, invece, le accelerazioni lineari sono di entità maggiore e rendono inaffidabile la stima ottenuta. Una stima effettuata con l'ausilio del solo accelerometro presentava valori sovrastimati, in quanto le accelerazioni lineari del sensore durante le oscillazioni della camminata si sommano alle componenti delle accelerazioni di gravità. Allo stesso tempo, nel caso di grandi e veloci variazioni di angolo, la stima senza l'ausilio dei giroscopi diventa più lenta a causa della velocità massima della convergenza della stima dell'orientazione tramite accelerometri nel filtro IMU. La soluzione ideale consiste, quindi, nel trattare i dati dei giroscopi allo scopo di eliminare la variazione del bias.

6. FILTRI

6.1 *Filtri principali*

In termini di algoritmi di fusione dei dati IMU, il filtro complementare e il filtro di Kalman sono gli algoritmi più utilizzati. Ognuno ha i suoi vantaggi e i suoi svantaggi. Il filtro complementare utilizza un algoritmo relativamente semplice, che solo richiede un calcolo leggero e facile da implementare.

Al contrario, il filtro Kalman è un filtro iterativo ed ha complessità computazionale efficiente ma elevata.

6.2 *Filtro complementare*

Il filtro complementare sfrutta i dati raccolti dall'accelerometro e dal giroscopio. Gestisce contemporaneamente al suo interno filtri passa-basso e filtri passa-alto. L'accelerometro misura tutte le forze che lavorano sull'oggetto ed acquisisce molto più del semplice vettore di gravità. Ogni piccola forza che lavora sull'oggetto disturberà completamente la sua misurazione. I dati dell'accelerometro sono affidabili solo a lungo termine, quindi è necessario utilizzare il filtro passa-basso.

Una volta ottenuta la posizione angolare con i dati registrati dal giroscopio, è possibile notare che è molto facile ottenere una misurazione accurata, non sensibile alle forze esterne. Purtroppo, a causa dell'integrazione nel tempo, la misurazione tende a spostarsi (*drift*), senza tornare a zero quando il sistema torna alla sua posizione originale. I dati del giroscopio

sono affidabili solo a breve termine, poiché iniziano a “spostarsi” a lungo termine.

I filtri passa-basso filtrano i segnali con frequenza elevata nel caso delle vibrazioni dell'accelerometro e i filtri passa-alto filtrano i segnali con bassa frequenza, come ad esempio il *drift* del giroscopio. La fusione delle due stime restituisce una stima globale dell'orientamento. Il principio del filtro complementare è illustrato nella [Figura 17](#).

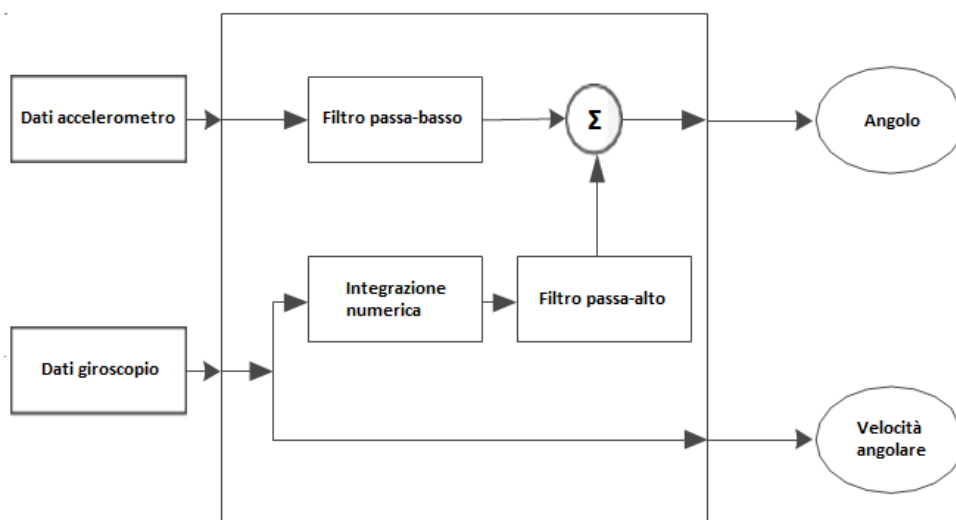


Figura 17: principio filtro complementare

La funzione del filtro passa-basso è quella di lasciar “passare” i cambiamenti a lungo termine, filtrando le oscillazioni a breve termine. Il filtro passa-alto fa la stessa cosa per oscillazioni a lungo termine, consentendo ai segnali di breve durata di “passare” filtrando i segnali che sono stabili nel tempo. Queste operazioni sono fondamentali per cancellare la deriva dei dati del giroscopio in modo di ottenere un angolo

di inclinazione stimato accurato. Il modello matematico del filtro complementare può essere rappresentato come segue (Equazione 4):

$$\theta_{Angolo} = \alpha * (\theta_{Angolo} + \omega_{Gyro} * dt) + (1 - \alpha) * a_{Acc}$$

Equazione 4: equazione del filtro complementare

θ_{Angolo} è l'angolo di inclinazione (beccheggio o rollio);

- α è il coefficiente di filtro;
- ω_{Gyro} rappresenta la velocità angolare dal giroscopio;
- a_{Acc} è l'angolo ottenuto attraverso i dati dall'accelerometro.

Il coefficiente di filtro α è determinato dall' Equazione 5

$$\alpha = \frac{\tau}{\tau + dt}$$

Equazione 5: coefficiente di filtro

dove τ è la costante di tempo del filtro.

La somma dei coefficienti di filtro, α e $1 - \alpha$, deve essere uguale a 1, in modo che l'output sia una stima accurata.

Per il filtro passa-basso, i segnali che sono molto più grandi della costante di tempo lasciano il filtro inalterato, mentre i segnali più piccoli della costante di tempo vengono filtrati. Per il filtro passa-alto vale il

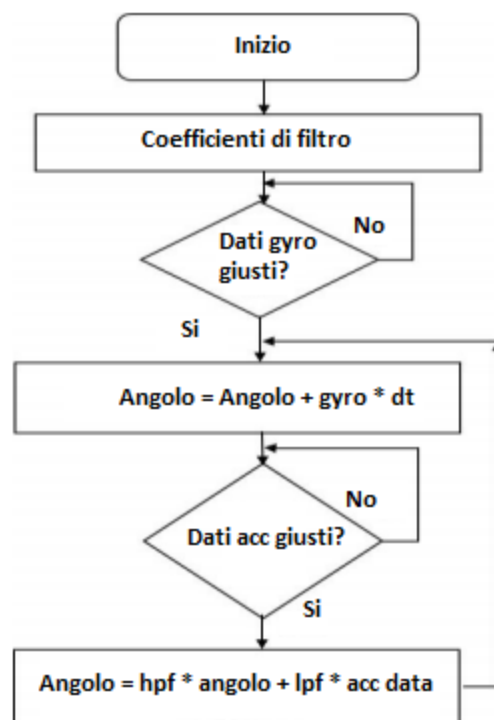


Figura 18: procedura algoritmo filtro complementare

ragionamento opposto. Per ogni intervallo di tempo, i dati del giroscopio vengono prima integrati con l'angolo corrente e poi combinati con i dati dell'accelerometro filtrati. La procedura di elaborazione dei dati del filtro complementare è mostrata nella [Figura 18](#).

Il filtro complementare per la fusione dei dati di un sensore IMU è implementato all'interno di un ciclo. I valori dell'angolo di Rollio e Beccheggio vengono aggiornati con l'uscita del giroscopio mediante l'integrazione nel tempo. Il filtro controlla quindi se l'entità della forza misurata dall'accelerometro ha un valore ragionevole di accelerazione gravitazionale. Se il valore è troppo piccolo o troppo grande, non verrà

preso in considerazione. In questo processo di fusione dei dati, l'angolo attuale di inclinazione è determinato dall'angolo di inclinazione precedente, dall'uscita attuale del giroscopio, dal coefficiente di filtro α e dall'uscita attuale dell'accelerometro.

7. DATA FUSION

7.1 *Data fusion traslazione verso destra e ritorno*

La prima cosa da fare all'interno dell'algoritmo di Data Fusion (Figura 19, Figura 20) è regolare i vari coefficienti di filtro.

Il primo parametro è quello relativo alla frequenza di aggiornamento del sensore IMU. Accelerometro e giroscopio del sensore LSM6DSM campionano alla frequenza di 208 Hz.

HPF e LPF sono i coefficienti di filtro α della Equazione 4. Per calcolare il coefficiente di filtro HPF si utilizza l'Equazione 5, utilizzando come $\tau = 15$ (tempo trascorso in secondi) e $dt = (1/208)$. HPF sarà uguale a 0.99967959.

Per calcolare il coefficiente di filtro LPF si utilizza l'Equazione 6:

$$LPF = \frac{dt}{\tau + dt}$$

Equazione 6: coefficiente di filtro LPF

utilizzando $\tau = 15$ (tempo trascorso in secondi) e $dt = (1/208)$ LPF sarà uguale a 0.00032041.

```
close all;clear; clc;

updateRate = 208;    % Hz
dt = 1/updateRate;  % s
hpf = 0.99967959;
lpf = 0.00032041;

acc_x = cell(size(sheet));
acc_y = acc_x;
acc_z = acc_x;
thetaX_acc = acc_x;
thetaY_acc = acc_x;
thetaZ_acc = acc_x;
gyr_x = acc_x;
gyr_y = acc_x;
gyr_z = acc_x;
thetaX = acc_x;
thetaY = acc_x;
thetaZ = acc_x;
for s = 1:numSheets
    [sheet{s},~,~] = xlsread(filename,s);
    updateRate = 208;    % Hz
    dt = 1/updateRate;  % s

    thisSheet = sheet{s};
```

Figura 19: algoritmo data fusion parte 1

```

acc_x{s} = thisSheet(:,3);
acc_y{s} = thisSheet(:,4);
acc_z{s} = thisSheet(:,5);
thetaX_acc{s} = lpf*atan2(thetaX_acc{s},sqrt((acc_z{s}*acc_z{s})+(acc_y{s}*acc_y{s})));
thetaY_acc{s} = lpf*atan2(thetaY_acc{s},sqrt((acc_z{s}*acc_z{s})+(acc_x{s}*acc_x{s})));
thetaZ_acc{s} = lpf*atan2(thetaZ_acc{s},sqrt((acc_y{s}*acc_y{s})+(acc_x{s}*acc_x{s})));

gyr_x{s} = thisSheet(:,6);
gyr_y{s} = thisSheet(:,7);
gyr_z{s} = thisSheet(:,8);
thetaX{s} = zeros(size(gyr_x{s}));
thetaY{s} = thetaX{s};
thetaZ{s} = thetaY{s};
for a = 1:length(gyr_x{s})
    if a == 1
        thetaX{s}(a) = hpf*thetaX{s}(a)*dt + thetaX_acc{s}(a);
        thetaY{s}(a) = hpf*thetaY{s}(a)*dt + thetaY_acc{s}(a);
        thetaZ{s}(a) = hpf*thetaZ{s}(a)*dt + thetaZ_acc{s}(a);
    else
        thetaX{s}(a) = hpf*(thetaX{s}(a-1) + gyr_x{s}(a)*dt) + thetaX_acc{s}(a);
        thetaY{s}(a) = hpf*(thetaY{s}(a-1) + gyr_y{s}(a)*dt) + thetaY_acc{s}(a);
        thetaZ{s}(a) = hpf*(thetaZ{s}(a-1) + gyr_z{s}(a)*dt) + thetaZ_acc{s}(a);
    end
end

```

Figura 20: algoritmo Data Fusion parte 2

Una volta inizializzati e riempiti gli array con i dati raccolti da accelerometro e giroscopio, l'algoritmo del filtro complementare incomincia ad operare.

I dati raccolti dall'accelerometro verranno trattati con le seguenti equazioni (Equazione 9, Equazione 8, Equazione 7):

$$\theta = \arctan\left(\frac{A_x}{\sqrt{A_y^2 + A_z^2}}\right)$$

Equazione 9: angolo beccheggio

$$\varphi = \arctan\left(\frac{A_y}{\sqrt{A_x^2 + A_z^2}}\right)$$

Equazione 8: angolo di rollio

$$\Phi = \arctan\left(\frac{A_z}{\sqrt{A_x^2 + A_y^2}}\right)$$

Equazione 7: angolo Φ rispetto all'orizzonte

I rispettivi risultati verranno moltiplicati al coefficiente di filtro LPF (Figura 21):

```
thetaX_acc{s} = lpf*atan2(thetaX_acc{s},sqrt((acc_z{s}*acc_z{s})+(acc_y{s}*acc_y{s})));
thetaY_acc{s} = lpf*atan2(thetaY_acc{s},sqrt((acc_z{s}*acc_z{s})+(acc_x{s}*acc_x{s})));
thetaZ_acc{s} = lpf*atan2(thetaZ_acc{s},sqrt((acc_y{s}*acc_y{s})+(acc_x{s}*acc_x{s})));
```

Figura 21: angolo di rollio, angolo beccheggio e angolo Φ rispetto all'orizzonte nel filtro complementare

I dati raccolti dal giroscopio vengono integrati con la costante di tempo e moltiplicati per il coefficiente di filtro HPF. Successivamente gli angoli ricavati mediante i filtri complementari vengono fusi (sommati) per restituire gli angoli di rollio, beccheggio e imbardata del sensore IMU (Figura 22).

```
for a = 1:length(gyr_x{s})
    if a == 1
        thetaX{s}(a) = hpf*thetaX{s}(a)*dt + thetaX_acc{s}(a);
        thetaY{s}(a) = hpf*thetaY{s}(a)*dt + thetaY_acc{s}(a);
        thetaZ{s}(a) = hpf*thetaZ{s}(a)*dt + thetaZ_acc{s}(a);
    else
        thetaX{s}(a) = hpf*(thetaX{s}(a-1) + gyr_x{s}(a)*dt) + thetaX_acc{s}(a);
        thetaY{s}(a) = hpf*(thetaY{s}(a-1) + gyr_y{s}(a)*dt) + thetaY_acc{s}(a);
        thetaZ{s}(a) = hpf*(thetaZ{s}(a-1) + gyr_z{s}(a)*dt) + thetaZ_acc{s}(a);
    end
end
```

Figura 22: calcolo angoli dai dati giroscopio e Data Fusion

Il risultato sarà l'inclinazione del sensore IMU sugli assi X, Y, Z.

L'output è illustrato in Figura 23:



Figura 23: output Data Fusion traslazione verso destra e ritorno

7.2 Data Fusion acquisizione 30 minuti

Per l'acquisizione di 30 minuti l'algoritmo è il medesimo.

La prima cosa da fare all'interno dell'algoritmo di Data Fusion (Figura 24, Figura 25) è regolare i vari coefficienti di filtro.

Il primo parametro è quello relativo alla frequenza di aggiornamento del sensore IMU. Accelerometro e giroscopio del sensore LSM6DSM campionano alla frequenza di 208 Hz.

HPF e LPF sono i coefficienti di filtro α della [Equazione 4](#). Per calcolare il coefficiente di filtro HPF si utilizza l'[Equazione 5](#), utilizzando come $\tau = 1800$ (tempo trascorso in secondi) e $dt = (1/208)$. HPF sarà uguale a 0,99999732906696296217157539643324.

Per calcolare il coefficiente di filtro LPF si utilizza l'[Equazione 6](#):

$$LPF = \frac{dt}{\tau + dt}$$

Equazione 10: coefficiente di filtro LPF

utilizzando $\tau = 1800$ (tempo trascorso in secondi) e $dt = (1/208)$ LPF sarà uguale a 0,00000267093303703782842460356676.

```
close all;clear; clc;

updateRate = 208;    % Hz
dt = 1/updateRate;  % s
hpf = 0,99999732906696296217157539643324;
lpf = 0,00000267093303703782842460356676;

acc_x = cell(size(sheet));
acc_y = acc_x;
acc_z = acc_x;
thetaX_acc = acc_x;
thetaY_acc = acc_x;
thetaZ_acc = acc_x;
gyr_x = acc_x;
gyr_y = acc_x;
gyr_z = acc_x;
thetaX = acc_x;
thetaY = acc_x;
thetaZ = acc_x;
for s = 1:numSheets
    [sheet{s},~,~] = xlsread(filename,s);
    updateRate = 208;    % Hz
    dt = 1/updateRate;  % s

    thisSheet = sheet{s};
```

Figura 24: algoritmo Data Fusion acquisizione 30 minuti parte 1

```

acc_x{s} = thisSheet(:,3);
acc_y{s} = thisSheet(:,4);
acc_z{s} = thisSheet(:,5);
thetaX_acc{s} = lpf*atan2(thetaX_acc{s},sqrt((acc_z{s}*acc_z{s})+(acc_y{s}*acc_y{s})));
thetaY_acc{s} = lpf*atan2(thetaY_acc{s},sqrt((acc_z{s}*acc_z{s})+(acc_x{s}*acc_x{s})));
thetaZ_acc{s} = lpf*atan2(thetaZ_acc{s},sqrt((acc_y{s}*acc_y{s})+(acc_x{s}*acc_x{s})));

gyr_x{s} = thisSheet(:,6);
gyr_y{s} = thisSheet(:,7);
gyr_z{s} = thisSheet(:,8);
thetaX{s} = zeros(size(gyr_x{s}));
thetaY{s} = thetaX{s};
thetaZ{s} = thetaY{s};
for a = 1:length(gyr_x{s})
    if a == 1
        thetaX{s}(a) = hpf*thetaX{s}(a)*dt + thetaX_acc{s}(a);
        thetaY{s}(a) = hpf*thetaY{s}(a)*dt + thetaY_acc{s}(a);
        thetaZ{s}(a) = hpf*thetaZ{s}(a)*dt + thetaZ_acc{s}(a);
    else
        thetaX{s}(a) = hpf*(thetaX{s}(a-1) + gyr_x{s}(a)*dt) + thetaX_acc{s}(a);
        thetaY{s}(a) = hpf*(thetaY{s}(a-1) + gyr_y{s}(a)*dt) + thetaY_acc{s}(a);
        thetaZ{s}(a) = hpf*(thetaZ{s}(a-1) + gyr_z{s}(a)*dt) + thetaZ_acc{s}(a);
    end
end
end

```

Figura 25: algoritmo Data Fusion acquisizione 30 minuti parte 2

Una volta inizializzati e riempiti gli array con i dati raccolti da accelerometro e giroscopio, l'algoritmo del filtro complementare incomincia ad operare.

I dati raccolti dall'accelerometro verranno trattati con le seguenti equazioni (Equazione 9, Equazione 8, Equazione 7):

$$\theta = \arctan\left(\frac{A_x}{\sqrt{A_y^2 + A_z^2}}\right)$$

Equazione 9: angolo beccheggio

$$\varphi = \arctan\left(\frac{A_y}{\sqrt{A_x^2 + A_z^2}}\right)$$

Equazione 8: angolo di rollio

$$\Phi = \arctan\left(\frac{A_z}{\sqrt{A_x^2 + A_y^2}}\right)$$

Equazione 7: angolo Φ rispetto all'orizzonte

I rispettivi risultati verranno moltiplicati al coefficiente di filtro LPF (Figura 26):

```
thetaX_acc{s} = lpf*atan2(thetaX_acc{s},sqrt((acc_z{s}*acc_z{s})+(acc_y{s}*acc_y{s})));
thetaY_acc{s} = lpf*atan2(thetaY_acc{s},sqrt((acc_z{s}*acc_z{s})+(acc_x{s}*acc_x{s})));
thetaZ_acc{s} = lpf*atan2(thetaZ_acc{s},sqrt((acc_y{s}*acc_y{s})+(acc_x{s}*acc_x{s})));
```

Figura 26: angolo di rollio, angolo beccheggio e angolo Φ rispetto all'orizzonte nel filtro complementare

I dati raccolti dal giroscopio vengono integrati con la costante di tempo e moltiplicati per il coefficiente di filtro HPF. Successivamente gli angoli ricavati mediante i filtri complementari vengono fusi (sommati) per restituire gli angoli di rollio, beccheggio e imbardata del sensore IMU (Figura 27).

```
for a = 1:length(gyr_x{s})
    if a == 1
        thetaX{s}(a) = hpf*thetaX{s}(a)*dt + thetaX_acc{s}(a);
        thetaY{s}(a) = hpf*thetaY{s}(a)*dt + thetaY_acc{s}(a);
        thetaZ{s}(a) = hpf*thetaZ{s}(a)*dt + thetaZ_acc{s}(a);
    else
        thetaX{s}(a) = hpf*(thetaX{s}(a-1) + gyr_x{s}(a)*dt) + thetaX_acc{s}(a);
        thetaY{s}(a) = hpf*(thetaY{s}(a-1) + gyr_y{s}(a)*dt) + thetaY_acc{s}(a);
        thetaZ{s}(a) = hpf*(thetaZ{s}(a-1) + gyr_z{s}(a)*dt) + thetaZ_acc{s}(a);
    end
end
```

Figura 27: calcolo angoli dai dati giroscopio e Data Fusion

Il risultato sarà l'inclinazione del sensore IMU sugli assi X, Y, Z.

L'output è illustrato in Figura 28:

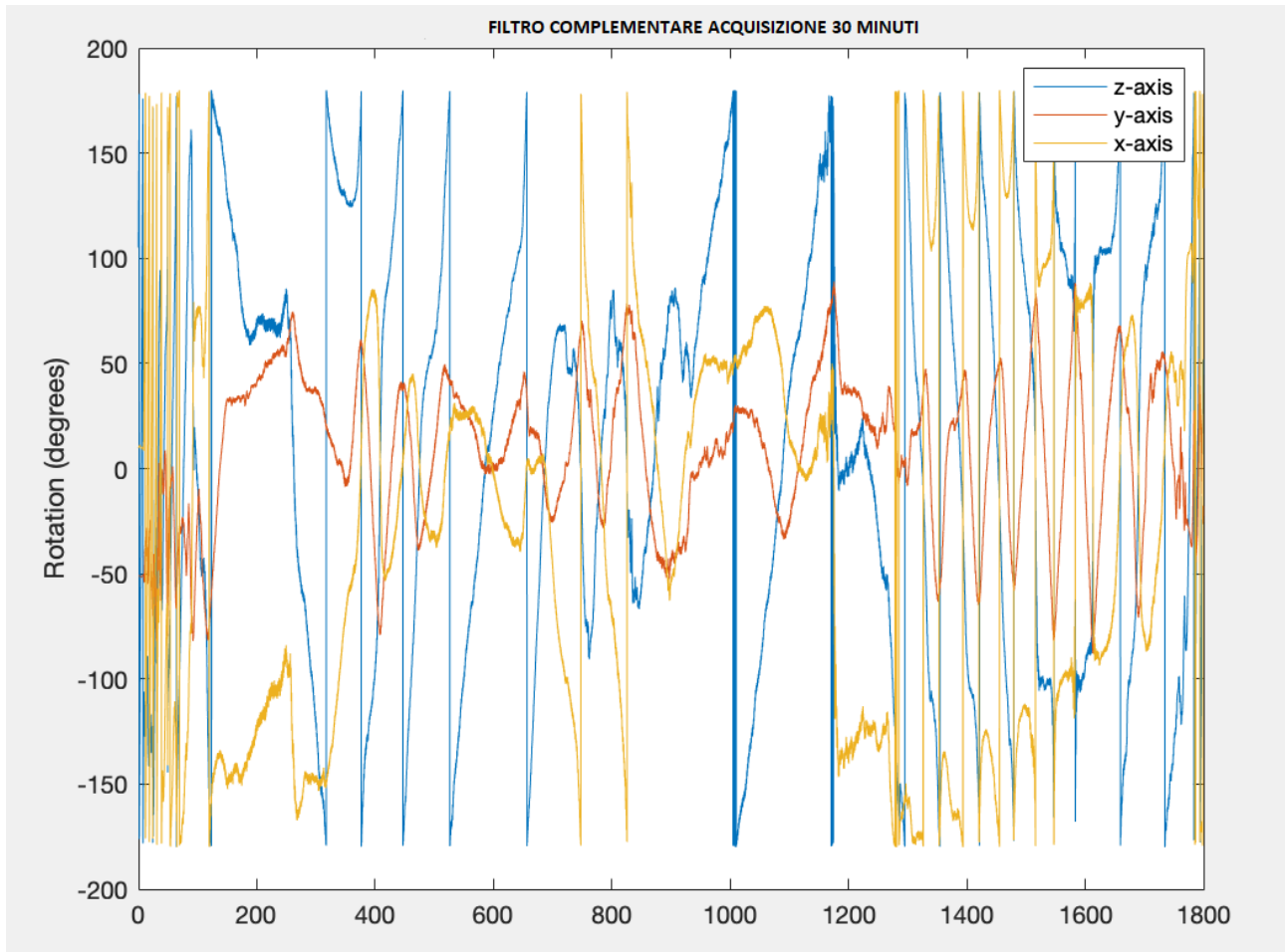


Figura 28: output algoritmo Data Fusion acquisizione 30 minuti

8. Conclusioni

Questa tesi si pone l'obiettivo di trovare un algoritmo di Data Fusion basato su filtri complementari e capace di calcolare in modo efficiente l'inclinazione dello stesso sensore.

Al fine di sviluppare questo algoritmo, sono stati raccolti dei dati mediante diversi sensori (magnetometro, giroscopio, accelerometro). In seguito, sono stati studiati i loro comportamenti in diverse situazioni. In primis è stata effettuata una traslazione dei sensori verso destra con conseguente ritorno in posizione iniziale. In seguito, è stata effettuata una camminata di 30 minuti attorno la facoltà di Ingegneria dell'Università Politecnica delle Marche. Tali dati sono stati elaborati tramite un'esplorazione algoritmica di data fusion.

In generale, è difficile ottenere un orientamento dell'inclinazione accurato usando solo l'accelerometro, perché il segnale derivato dalla funzione trigonometrica inversa è sensibile ai disturbi esterni. I dati del giroscopio sono necessari per compensare gli svantaggi dei dati dall'accelerometro. Poiché i dati del giroscopio soffrono di deriva a lungo termine, sono necessari algoritmi di filtro per risolvere tali problemi.

I risultati mettono in evidenza che in poche righe di codice è possibile ottenere un algoritmo di Data Fusion basato su filtri complementari e capace di calcolare in modo efficiente l'inclinazione dello stesso sensore.

Tra i filtri, quello complementare è la tipologia più vantaggiosa da adottare in quanto richiede pochi calcoli e solamente due coefficienti di filtro da

regolare, il che riduce notevolmente il carico di lavoro e la complessità computazionale.

In conclusione, l'algoritmo è quindi meno sensibile alle variazioni. Usando la giusta messa a punto dei coefficienti di filtro, il risultato del filtro complementare può essere molto stabile e preciso.

Riferimenti

Bernardo. (2011). Tratto da <https://www.settorezero.com/wordpress/cosa-sono-come-funzionano-e-a-cosa-servono-gli-accelerometri/>

Casolino. (2016). Tratto da <https://www.scientificast.it/funziona-magnetometro-dello-smartphone/>

Mathworks. (2020). Tratto da <https://it.mathworks.com/products/matlab.html>

Pinto. (2017). Tratto da <https://www.thedifferentgroup.com/2017/02/10/giroscopio-come-funziona/>

Indice delle equazioni

Equazione 1: momento angolare	14
Equazione 2: momento della forza.....	14
Equazione 3: $\tau = 0$	14
Equazione 4: equazione del filtro complementare	33
Equazione 5: coefficiente di filtro.....	33
Equazione 6: coefficiente di filtro LPF	36
Equazione 7: angolo Φ rispetto all'orizzonte	38
Equazione 8: angolo di rollio	38
Equazione 9: angolo beccheggio	38
Equazione 10: coefficiente di filtro LPF	41

Indice delle figure

Figura 1: logo di Matlab.....	8
Figura 2: piattaforma IMU.....	10
Figura 3: esempio accelerometro.....	12
Figura 4: funzionamento giroscopio (Pinto, 2017).....	13
Figura 5: sensore LSM6DSM.....	16
Figura 6: versi degli assi per le accelerazioni lineari.....	17
Figura 7: sensi di rotazione degli assi per le velocità angolari.....	17
Figura 8: angoli di rollio (colore rosso), beccheggio (viola) e imbardata (giallo).....	18
Figura 9: raccolta dati accelerometro.....	22
Figura 10: raccolta dati giroscopio.....	22
Figura 11: raccolta dati magnetometro.....	22
Figura 12: algoritmo di partenza traslazione verso destra e ritorno.....	24
Figura 13: output algoritmo traslazione verso destra e ritorno.....	25
Figura 14: percorso e rappresentazione grafica dell'andamento.....	26
Figura 15: algoritmo acquisizione 30 minuti.....	29
Figura 16: output algoritmo acquisizione 30 minuti.....	29
Figura 17: principio filtro complementare.....	32
Figura 18: procedura algoritmo filtro complementare.....	34
Figura 19: algoritmo data fusion parte 1.....	37
Figura 20: algoritmo Data Fusion parte 2.....	38
Figura 21: angolo di rollio, angolo beccheggio e angolo Φ rispetto all'orizzonte nel filtro complementare..	39
Figura 22: calcolo angoli dai dati giroscopio e Data Fusion.....	39
Figura 23: output Data Fusion traslazione verso destra e ritorno.....	40
Figura 24: algoritmo Data Fusion acquisizione 30 minuti parte 1.....	42
Figura 25: algoritmo Data Fusion acquisizione 30 minuti parte 2.....	43
Figura 26: angolo di rollio, angolo beccheggio e angolo Φ rispetto all'orizzonte nel filtro complementare..	44
Figura 27: calcolo angoli dai dati giroscopio e Data Fusion.....	44
Figura 28: output algoritmo Data Fusion acquisizione 30 minuti.....	45