

ALMA MATER STUDIORUM
UNIVERSITA' DI BOLOGNA

SCUOLA DI INGEGNERIA E ARCHITETTURA
Sede di Forlì

Corso di Laurea in
INGEGNERIA AEROSPAZIALE
Classe L-9

ELABORATO FINALE DI LAUREA

in Elaborazione Dati per la Navigazione

Progetto e analisi di algoritmi per la Pedestrian Navigation

CANDIDATO
Francesco Belli

RELATORE
Prof. Matteo Zanzi

Anno Accademico 2016/2017

Sommario

Indice delle Figure	2
Ringraziamenti	3
1 Introduzione.....	4
2 INS – Inertial Navigation System.....	6
2.1 Descrizione Generale	6
2.2 Accelerometro	7
2.3 Accelerometri MEMS	12
2.4 Giroscopio	13
2.5 Errori nei sensori MEMS	15
3 Modellizzazione e Simulazione della Camminata.....	18
3.1 Ricerca di modelli esistenti	18
3.2 Progetto di un modello cinematico.....	19
3.3 Implementazione in MATLAB	22
3.4 Progettazione del filtro	25
3.5 Calcolo dell’assetto (angolo γ).....	28
4 Algoritmi	30
4.1 Integrazione.....	30
4.2 Applicazione “Sensor Kinetics”.....	33
4.3 Contapassi	35
4.4 Progetto OpenShoe.....	36
4.5 Impostazioni iniziali.....	38
4.6 Modello ISU del Sistema e Osservatore dello Stato	40
4.7 Zero-Velocity Detector	46
4.8 Funzionamento dell’INS con Zero-Velocity Update	49
4.9 Grafici e Valutazione dell’Algoritmo	57
5 Analisi dei Dati.....	61
5.1 Obiettivo dell’Analisi.....	61
5.2 Trasformata di Fourier	61
5.3 La funzione Periodogram in MATLAB	64
5.4 Analisi dei Dati IMU.....	65
5.5 Rilevamento del Cambio di Orientamento.....	68
Conclusioni.....	73
Riferimenti.....	74

Indice delle Figure

Figura 1: Sistema di riferimento ECI	6
Figura 2: Schema di un accelerometro	7
Figura 3: Sistemi di riferimento NED e body	10
Figura 4: Accelerometro MEMS	12
Figura 5: Giroscopio meccanico e giroscopio ottico	13
Figura 6: Giroscopio MEMS	15
Figura 7: Modello dinamico di camminata	18
Figura 8: Modello cinematico della camminata	19
Figura 9: Andamento degli angoli per due cicli	21
Figura 10: Sistemi di riferimento	23
Figura 11: Accelerazioni misurate in assi body	24
Figura 12: Calibrazione di τ in Simulink.....	26
Figura 13: Accelerazioni ottenute con l'applicazione del filtro.....	27
Figura 14: Andamento dei diversi γ	29
Figura 15: Ciclo della camminata.....	31
Figura 16: Velocità lungo x tramite integrazione.....	32
Figura 17: Posizione x tramite integrazione	32
Figura 18: Schermata principale dell'applicazione Sensor Kinetics	33
Figura 19: Grafici delle accelerazioni e degli angoli di assetto durante gli 8 passi .	34
Figura 20: Andamento del modulo dell'accelerazione	35
Figura 21: Piattaforma IMU del progetto OpenShoe	37
Figura 22: Funzionamento del sistema inerziale e delle integrazioni effettuate	37
Figura 23: Schema dell'INS integrato con ZUPT detector.....	38
Figura 24: Modello della Terra WGS84.....	39
Figura 25: Sistema e Osservatore dello stato	45
Figura 26: Dati dalla piattaforma IMU.....	57
Figura 27: Velocità calcolate lungo i tre assi	58
Figura 28: Andamento della coordinata verticale, della velocità totale e ZUPT	58
Figura 29: Andamento dell'assetto (angoli di Eulero).....	59
Figura 30: Traiettoria nel piano	59
Figura 31: Spettro di un segnale campionato	62
Figura 32: PSD degli accelerometri singoli e del modulo dell'accelerazione	66
Figura 33: PSD dei singoli giroscopio e del modulo della velocità angolare	66
Figura 34: Fenomeno dell'aliasing	68
Figura 35: Confronto tra le PSD dei tre giroscopi.....	69
Figura 36: Confronto tra le PSD del percorso completo e del tratto rettilineo per i giroscopi	70
Figura 37: Segnale del giroscopio di imbardata filtrato	72

Ringraziamenti

Prima di esporre il lavoro svolto, vorrei ringraziare innanzitutto il professore Matteo Zanzi, che mi ha seguito pazientemente durante la ricerca e in tutte le fasi di elaborazione della tesi, aiutandomi con la sua esperienza in diverse circostanze.

Devo anche ringraziare profondamente la mia famiglia e i miei amici, che mi hanno incoraggiato durante l'intero percorso di studio, permettendomi di superare tutti i momenti di difficoltà e giungere così a questo primo traguardo.

1 Introduzione

Al giorno d'oggi risulta particolarmente comune l'utilizzo del segnale GPS per determinare con accuratezza la posizione e la velocità di un certo sistema, nonché in ambito di navigazione e calcolo della distanza percorsa. Una problematica che questa metodologia presenta è la necessità di una minima ricezione del segnale, motivo per cui è opportuno sfruttare altre tecniche in tutte quelle circostanze in cui è assente, ad esempio all'interno di tunnel, magazzini, ospedali. Basti pensare a una situazione di emergenza come l'incendio all'interno di un grande edificio: le squadre che entrano in azione per portare in salvo tutte le persone rischiano di perdersi, oppure potrebbero non essere organizzate nella maniera più ottimale per completare la missione nel minor tempo possibile proprio a causa della mancanza di un adeguato sistema di localizzazione.

Una possibile soluzione prevede l'utilizzo di una rete Wi-Fi o altri tipi di segnali, ma ciò richiede comunque l'installazione di apparecchiature che possono essere costose e la possibilità da parte degli utenti di interagire con esse. Da qualche anno, è cresciuta la fiducia nel potenziale di sensori inerziali, come accelerometri e giroscopi, per poter riuscire a stimare il percorso effettuato, o almeno la distanza percorsa a partire dai dati che i sensori stessi forniscono. Il tutto deve essere supportato però da particolari accorgimenti di calcolo e di elaborazione dei segnali, per compensare difetti che, come verrà illustrato, tali sensori presentano e che causano l'inutilizzabilità dei risultati già dopo una decina di secondi. In particolare, oltre ai dati ottenuti dalla piattaforma IMU (Inertial Measurement Unit) e alle equazioni di calcolo standard, viene introdotta un'importante proprietà della camminata che caratterizza la sua periodicità: per certi istanti ad ogni ciclo il piede risulta stazionario, dunque possiede velocità nulla, ed è possibile sfruttare quest'informazione per correggere gli errori che si accumulano durante le integrazioni.

L'attività svolta, su cui si basa questa tesi, si è incentrata innanzitutto sull'ideazione e successiva implementazione in MATLAB di un modello di camminata semplificato in 2D, per poi simulare l'acquisizione di dati da accelerometri e giroscopi e successivamente calcolare la distanza percorsa. In seguito, si è tentato di sfruttare l'algoritmo per arrivare allo stesso risultato partendo da dati reali, acquisiti dai sensori presenti nel cellulare e raccolti tramite

un'applicazione già esistente. Poiché non si è raggiunto un risultato soddisfacente, si è poi pensato di studiare e provare un codice MATLAB (oltre a tutti gli aspetti teorici che ne sono le fondamenta) messo a disposizione online da un gruppo di ricercatori del KTH relativamente al progetto OpenShoe, che ha compiuto ottimi progressi nell'ambito della "pedestrian navigation". Infine, è stata fatta un'analisi dei dati IMU acquisiti sperimentalmente in relazione al progetto già citato (anch'essi disponibili online), per effettuare delle verifiche sugli algoritmi progettati in precedenza e per determinare la possibilità o meno di rilevare cambi di direzione dai soli segnali dei sensori.

2 INS – Inertial Navigation System

2.1 Descrizione Generale

Il sistema di navigazione inerziale è costituito da un insieme di sensori, tra cui accelerometri e giroscopi, che permettono di ottenere misure di accelerazione e velocità angolare rispetto ad un sistema di riferimento inerziale, come ad esempio, entro periodi di tempo limitati, il sistema ECI (Earth Centered Inertial), che possiede l'origine nel centro della Terra, l'asse x rivolto al punto gamma della costellazione dell'Ariete, l'asse z parallelo all'asse di rotazione, e l'asse y di conseguenza per ottenere una terna destrorsa. Per definizione, un sistema di riferimento è inerziale se un corpo accelera nel solo caso in cui la risultante delle forze che agiscono su di esso sia diversa da zero.

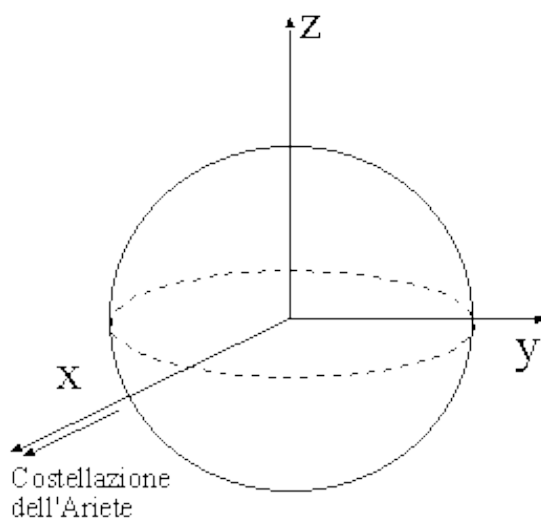


Figura 1: Sistema di riferimento ECI

Queste misure vengono poi utilizzate per diversi scopi, ad esempio per stimare la posizione, la velocità e l'assetto attraverso equazioni specifiche implementate da un calcolatore che raccoglie i dati. Il vantaggio di questi sensori è che forniscono le misure senza sfruttare riferimenti o segnali esterni, come accade invece con il GPS; presentano però diverse tipologie di errori che ne

impediscono l'utilizzo prolungato senza le opportune correzioni. In riferimento al lavoro di tesi, verranno descritte in seguito le principali caratteristiche degli accelerometri e dei giroscopi: principi di funzionamento, alcuni esempi, e tipologie di errori.

2.2 Accelerometro

Un accelerometro è costituito da una massa di prova m che può spostarsi lungo quello che viene chiamato "asse di ingresso" dell'accelerometro (in figura 2 l'asse x), a seguito dell'applicazione di forze esterne. In particolare, la massa di prova è contenuta all'interno di un case e collegata ad esso tramite una molla di costante elastica k , ed uno smorzatore di coefficiente di attrito viscoso b , che esercitano rispettivamente una forza di richiamo e una di attrito viscoso sulla massa di prova. In situazioni di forza esterna lungo l'asse di ingresso nulla, la massa è allineata col valore di riferimento x_0 , mentre quando vengono applicate sollecitazioni, essa si sposta insieme al cursore (x_m) che indica l'entità dello spostamento, ed il segno, rispetto al riferimento iniziale.

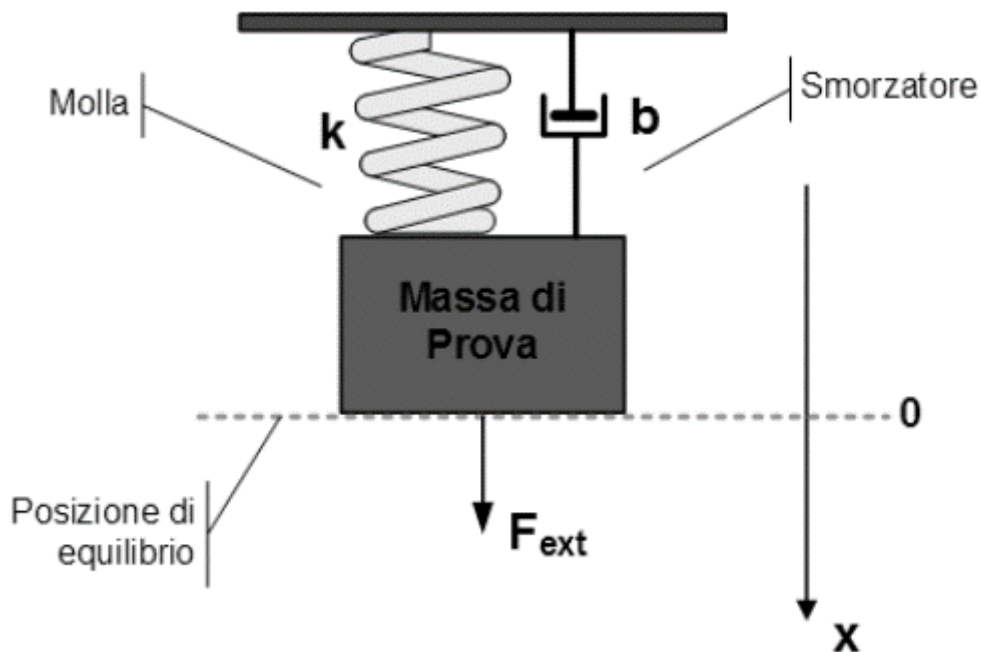


Figura 2: Schema di un accelerometro

Ipotizzando che la direzione longitudinale dell'accelerometro (ovvero il suo asse di ingresso) sia inclinata di un generico angolo α rispetto al piano orizzontale, positivo in senso orario, e che venga applicata un'accelerazione costante lungo lo stesso asse di ingresso, allora la massa comincerà a spostarsi e ad oscillare, sino a stabilizzarsi in una posizione di equilibrio che evidenzierà un discostamento tra x_o e x_m . Il modello matematico che descrive il moto della massa di prova, applicando il secondo principio della dinamica, risulta [1]:

$$m\ddot{x}_m = -k(x_m - x_o) - b(\dot{x}_m - \dot{x}_o) - mg \sin \alpha \quad (2.1)$$

Ponendo:

$$\begin{cases} d = x_m - x_o \\ \dot{d} = \dot{x}_m - \dot{x}_o \\ \ddot{d} = \ddot{x}_m - \ddot{x}_o = \ddot{x}_m - a \end{cases}$$

Si ottiene:

$$m(\ddot{d} + a) = -kd - b\dot{d} - mg \sin \alpha \quad (2.2)$$

Facendo il rapporto per m ad entrambi i membri, portando la variabile d al primo membro e considerando il sistema a regime (equilibrio a seguito delle oscillazioni), ovvero $\ddot{d} = \dot{d} = 0$, l'equazione diventa:

$$\frac{k}{m}d = -a - g \sin \alpha \quad (2.3)$$

Riscrivendola rispetto all'accelerazione a :

$$a = -\frac{k}{m}d - g \sin \alpha \quad (2.4)$$

Definendo quella che è l'accelerazione misurata, rispetto alla distanza d :

$$a_m = -\frac{k}{m}d$$

Si ottiene infine:

$$a = a_m - g \sin \alpha \quad (2.5)$$

Si nota dunque che per conoscere l'accelerazione lineare cui è sottoposto il sistema, è necessario conoscere l'angolo di inclinazione, e sottrarre il termine $g \sin \alpha$ all'accelerazione misurata, che rappresenta la componente dell'accelerazione di gravità lungo l'asse di ingresso dell'accelerometro.

Si nota che, nel caso in cui l'accelerazione a sia pari a zero (ovvero che il sistema sia fermo), allora in quel caso a partire dall'accelerazione misurata è possibile calcolare l'angolo di cui è inclinato l'asse di ingresso dell'accelerometro, che viene dunque sfruttato come inclinometro:

$$\alpha = \sin^{-1} \frac{a_m}{g} \quad (2.6)$$

Si può estendere questa proprietà al caso tridimensionale, avendo a disposizione un accelerometro triassiale, costituito da tre accelerometri i cui assi di ingresso sono disposti come tre assi ortogonali. L'obiettivo è quello di trovare il valore degli angoli di cui il sistema costituito da questi accelerometri è inclinato rispetto a quello di navigazione, di assi Nord, Est e Down (NED), descritto in figura 3. Definendo i seguenti vettori

\mathbf{a} vettore accelerazione

\mathbf{a}_m vettore delle tre misure ottenute dai tre accelerometri

\mathbf{g} vettore accelerazione di gravità

vale la seguente relazione:

$$\mathbf{a} = \mathbf{a}_m + \mathbf{g} \quad (2.7)$$

Per poter scrivere la (2.7) nel sistema di riferimento assi corpo (body frame), ovvero solidale alla piattaforma degli accelerometri (ed in generale al sistema inerziale utilizzato), è necessario applicare al vettore dell'accelerazione gravitazionale una rotazione tramite una matrice che permetta di passare dal sistema NED (North, East, Down) al sistema body, attraverso i tre angoli di Eulero:

- ψ angolo di prua (heading), attorno all'asse di imbardata
- ϑ angolo di elevazione (elevation), attorno all'asse di beccheggio
- φ angolo di banco (bank angle), attorno all'asse di rollio

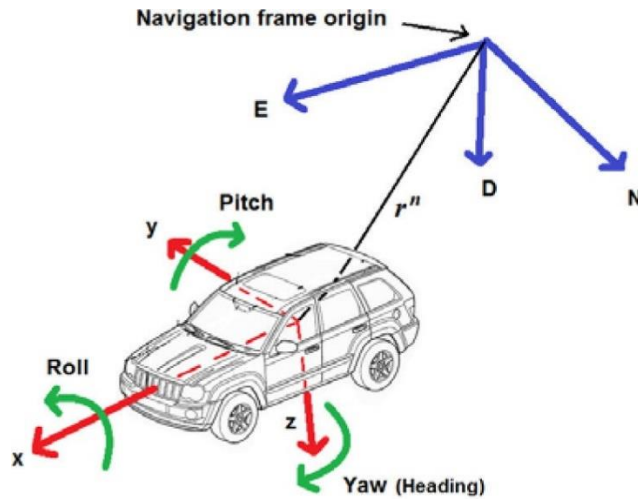


Figura 3: Sistemi di riferimento NED (blu) e body (rosso)

Definiti i seguenti angoli, e applicando le rotazioni in ordine attorno agli assi di imbardata, beccheggio e rollio, si ottiene la seguente matrice di rotazione complessiva dal sistema assi NED al sistema assi body (si intende $c=\cos$ e $s=\sin$):

$$C_{NED}^{body} = \begin{pmatrix} c \vartheta c \psi & c \vartheta s \psi & -s \vartheta \\ s \vartheta s \varphi c \psi - c \varphi s \psi & s \vartheta s \varphi s \psi + c \varphi c \psi & s \varphi c \vartheta \\ s \vartheta c \varphi c \psi + s \varphi s \psi & s \vartheta c \varphi s \psi - s \varphi c \psi & c \varphi c \vartheta \end{pmatrix} \quad (2.8)$$

Essendo poi la matrice di rotazione una matrice ortogonale, essa è invertibile,

e l'inversa coincide con la sua trasposta, da cui è facile capire che

$$C_{body}^{NED} = (C_{NED}^{body})^T \quad (2.9)$$

Considerando che il vettore dell'accelerazione gravitazionale in assi NED risulta

$$\mathbf{g}^{NED} = \begin{pmatrix} 0 \\ 0 \\ -g \end{pmatrix}$$

e applicando ad esso la rotazione di matrice (2.8), per poi sostituirlo nella (2.7) scritta in assi body (assi x, y e z), si ottiene:

$$\begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix} = \begin{pmatrix} a_{mx} \\ a_{my} \\ a_{mz} \end{pmatrix} + \begin{pmatrix} -g \sin \vartheta \\ g \sin \varphi \cos \vartheta \\ g \cos \varphi \cos \vartheta \end{pmatrix} \quad (2.10)$$

Per utilizzare l'accelerometro triassiale come inclinometro nel caso tridimensionale, deve valere l'ipotesi $\mathbf{a} = \mathbf{0}$, ovvero i sensori sono fermi e agisce la sola forza gravitazionale. Scrivendo le tre equazioni in un sistema e semplificando, si ottengono le seguenti relazioni per calcolare gli angoli con cui il sistema di riferimento body è inclinato rispetto al NED:

$$\begin{cases} \vartheta = \sin^{-1} \frac{a_{mx}}{g} \\ \varphi = \arctan2(a_{my}, a_{mz}) \end{cases} \quad (2.11)$$

2.3 Accelerometri MEMS

In generale i MEMS (Micro Electro-Mechanical Systems) sono dispositivi i cui componenti hanno dimensioni che si aggirano intorno all'ordine di grandezza del micrometro: si parla di microsensori e di microprocessore. I sensori sono di tipo analogico, e solitamente reagiscono a sollecitazioni di tipo meccanico (ma anche biologico, ottico, termico e magnetico), per poi comunicare con il circuito elettrico integrato tramite una tensione variabile in uscita e permettere al processore di elaborare l'informazione traducendola nella misura di grandezza desiderata. Il materiale con cui generalmente vengono realizzati è il silicio, e lo sviluppo notevole raggiunto con esso nei microprocessori utilizzati nei computer ha permesso ai MEMS di diffondersi anche per il costo contenuto.

Nel caso dell'accelerometro, esso può essere di tipo resistivo o capacitivo. L'accelerometro resistivo è costituito da una massa metallica centrale tenuta sospesa da due lamierini (due estensimetri), elementi conduttivi che a seguito di una modifica della loro lunghezza causano una variazione della resistenza. Se il sistema subisce un'accelerazione, la massa sospesa si disallineerà e i lamierini modificheranno la propria resistenza a seguito dell'allungamento, causando una variazione di tensione in uscita misurabile tramite un voltmetro. L'accelerometro capacitivo è composto da un condensatore, di cui un'armatura è fissa, mentre l'altra è sospesa da una membrana elastica dielettrica. In caso di sollecitazione esterna, l'armatura mobile si sposterà causando una variazione della capacità, che verrà misurata e trasformata in un valore di accelerazione [2].

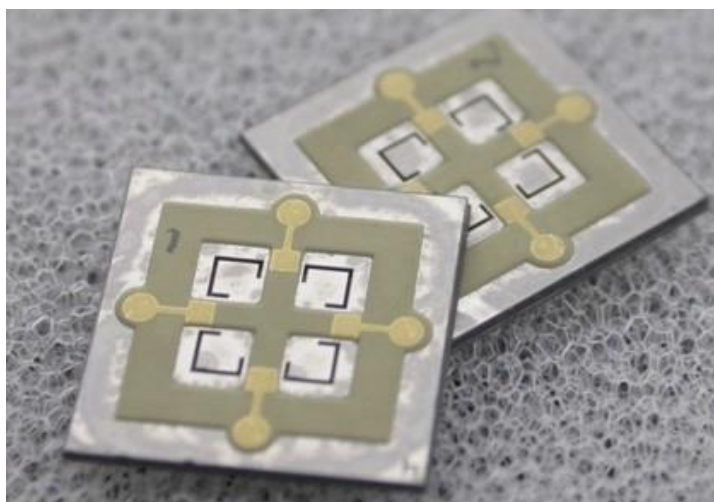


Figura 4: Accelerometro MEMS

2.4 Giroscopio

Il giroscopio è un sensore che ci permette di misurare la velocità angolare attorno ad un certo asse, in modo da poter calcolare tramite essa la variazione di angolo subita dalla piattaforma attorno allo stesso asse e aggiornare la sua inclinazione rispetto al sistema di riferimento di navigazione. Esistono principalmente tre tipologie di giroscopi, che sfruttano diversi principi fisici:

- Giroscopi meccanici
- Giroscopi ottici
- Giroscopi MEMS

Il giroscopio meccanico (figura 5) è costituito da un rotore che possiede una certa velocità angolare attorno al proprio asse di rotazione (o di spin), mantenuto all'interno di una sospensione cardanica. Avendo un elevato momento angolare, dato dal prodotto tra il proprio momento d'inerzia e la velocità angolare con cui è posto in rotazione, il rotore tenderà a mantenere il suo orientamento iniziale, portando ad uno sfasamento dell'angolo di uno dei cardini della sospensione. In particolare, oltre all'asse di spin, si definisce anche l'asse di input quello relativo al momento esterno applicato, e l'asse di output quello attorno a cui ruota il giroscopio, che risulta ortogonale sia all'asse di spin che a quello di output.

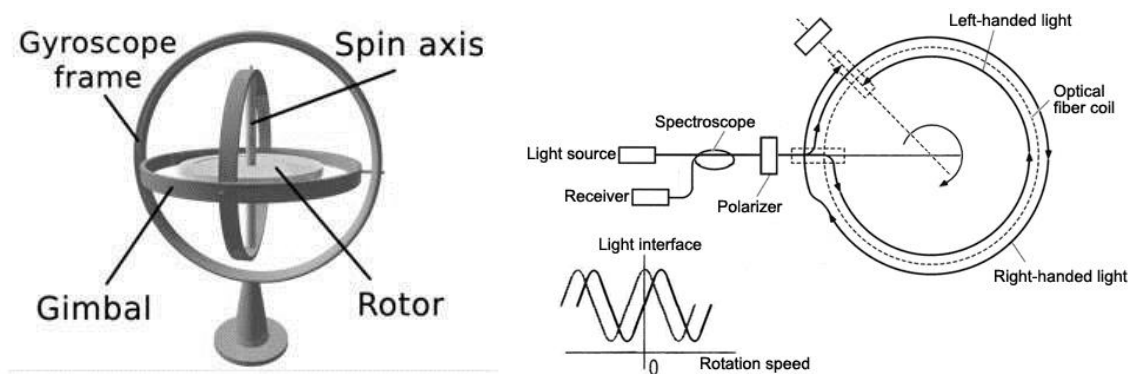


Figura 5: Giroscopio meccanico (sinistra) e giroscopio ottico (destra)

Il giroscopio ottico (FOG: Fibre Optic Gyroscope) misura la velocità angolare sfruttando l'effetto Sagnac: all'interno di una fibra ottica vengono inviati due raggi di luce in direzione opposta, di modo che nel momento in cui il sensore è posto in rotazione, il completamento del giro avverrà in maniera sfasata, poiché il raggio che viaggia in verso opposto alla rotazione impiegherà minor tempo. Da questo sfasamento è possibile risalire alla misura della velocità angolare, che risulta maggiormente precisa se la fibra ottica è più lunga, il che può risultare svantaggioso in termini di dimensioni del sensore.

Il tipo di giroscopio che risulta più comunemente utilizzato, e che risulta più di interesse per questa tesi, è sicuramente quello MEMS, legato alla stessa tecnologia illustrata nel caso dell'accelerometro. Il principio fisico su cui si basa questo tipo di sensore è la Forza di Coriolis:

$$\vec{F}_c = -2m\vec{\omega} \times \vec{v} \quad (2.12)$$

Quando si ha una massa che si muove con una certa velocità \vec{v} ed il sistema di cui fa parte è sottoposto ad una velocità angolare $\vec{\omega}$, allora nasce una forza che risulta ortogonale ad entrambi i vettori. Nel caso del giroscopio si ha una configurazione "a forchetta", in cui due masse vengono fatte oscillare una in verso opposto all'altra lungo una certa direzione, cosicché alla presenza di una velocità angolare nasce una coppia di forze come illustrato in figura 6, per cui le masse cominciano ad oscillare anche lungo la direzione ortogonale. Da questa seconda vibrazione deriva una variazione della capacità elettrica, proporzionale alla velocità angolare con cui ruota il sistema, che può essere dunque calcolata. Nonostante la precisione di questi giroscopi non raggiunga quella dei FOG, posseggono diversi vantaggi, tra cui dimensioni e peso ridotti, basso consumo di potenza, semplice costruzione e basso costo.

Un'altra modalità con cui si calcola la velocità angolare prevede l'impiego di un sensore di deformazione (strain gage), che riesca a valutare la sollecitazione causata dalla coppia torcente di Coriolis alla base del giroscopio a forchetta, in cui è presente una giunzione elastica che reagisce con una coppia resistente.

Come nel caso degli accelerometri, si possono misurare le velocità angolari lungo tre assi ortogonali tramite un giroscopio triassiale, composto da tre giroscopi i cui assi di ingresso sono ortogonali tra di loro.

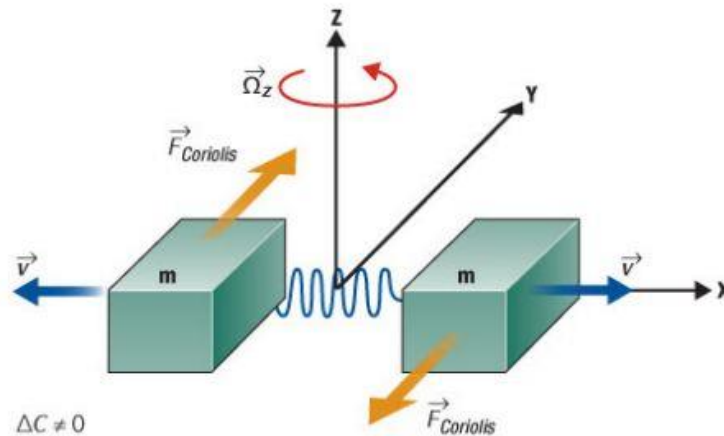


Figura 6: Giroscopio MEMS

2.5 Errori nei sensori MEMS

Le misure che questi sensori forniscono sono affette da diverse tipologie di errori, ed è importante conoscerne la natura per capire come compensare la loro influenza nel momento in cui si utilizzano le grandezze misurate per effettuare dei calcoli. Innanzitutto, è bene distinguere tra queste due categorie:

- Errori deterministici: sono dovuti ad aspetti costruttivi, legati anche alla tecnologia utilizzata e alla temperatura di funzionamento del sensore
- Errori stocastici: sono disturbi legati a processi random

Per entrare più nel dettaglio, distinguiamo tra fattore di scala, rumore bianco ed errore costante (deriva o bias).

Il fattore di scala è sostanzialmente una relazione che intercorre tra l'input e l'output del sensore, per cui per ottenere la misurazione corretta è necessario compensare moltiplicando per un certo fattore (o matrice nel caso la grandezza sia un vettore). Questo tipo di disturbo è maggiormente influente nel caso di brusche accelerazioni o rotazioni.

Il rumore termo-meccanico gaussiano bianco varia molto più velocemente rispetto al periodo di campionamento del sensore (agisce a frequenze più elevate). Può essere pensato come una successione molto rapida di valori aleatori (random) senza alcuna correlazione tra di loro, a media nulla e distribuita con varianza σ^2 . Indicando con \mathbf{w} il vettore aleatorio che contiene gli errori, vale dunque:

$$\begin{aligned} E[\mathbf{w}] &= 0 \\ E[\mathbf{w}\mathbf{w}^T] &= \sigma^2 I \end{aligned}$$

Dove $E[.]$ rappresenta il valore atteso, ovvero la media, e I è la matrice identità. Per compensare questo tipo di errore nel simulatore realizzato, è stato applicato un filtro passa-basso, come verrà illustrato più nel dettaglio in seguito.

L'errore costante viene chiamato bias nel caso degli accelerometri e deriva (drift) per i giroscopi: rappresenta il valor medio dell'uscita del sensore quando l'accelerometro (o il giroscopio) è in realtà fermo (con velocità angolare nulla). Esso è legato principalmente a difetti costruttivi. Il modello matematico per le misure degli accelerometri diventa [3]:

$$\mathbf{a}_m = \mathbf{a} - \mathbf{g} + \mathbf{b} \quad (2.13)$$

in cui \mathbf{a}_m è il vettore delle misure ottenute da un accelerometro triassiale, e \mathbf{b} è il vettore dei tre bias che agisce sulla terna di sensori. Allo stesso modo, si può scrivere per i giroscopi:

$$\boldsymbol{\omega}_m = \boldsymbol{\omega} + \mathbf{d} \quad (2.14)$$

dove analogamente $\boldsymbol{\omega}_m$ è il vettore delle misure ottenute da un giroscopio triassiale, e \mathbf{d} il vettore delle tre derive. Il problema principale per questo tipo di disturbi è legato all'integrazione che si effettua per il calcolo delle variazioni di angoli dalle misure dei giroscopi e per il calcolo della velocità e posizione dai dati forniti dagli accelerometri.

Definendo $\vartheta_m(t)$ l'angolo misurato e $\vartheta(t)$ l'angolo reale, è facile ottenere il

seguinte risultato:

$$\vartheta_m(t) = \int_0^t \omega_m(\tau) d\tau = \int_0^t (\omega(\tau) + d) d\tau = \vartheta(t) + td \quad (2.15)$$

Si nota che l'errore dovuto alla deriva nella misura dell'angolo cresce linearmente col tempo, portando ad un risultato inaffidabile.

La situazione è ancora più critica nel caso si voglia ottenere la posizione partendo dalla misura di accelerazione. Attraverso simili passaggi si ottiene infatti:

$$v_m(t) = v(t) + bt \quad (2.16)$$

$$p_m(t) = p(t) + \frac{1}{2}bt^2 \quad (2.17)$$

Poiché l'errore influenza parabolicamente la misura della posizione, questa diventa presto inattendibile.

In realtà è possibile compensare almeno in parte l'errore di bias o deriva, misurando il valore ottenuto dal sensore quando ci si aspetterebbe un dato nullo e correggendo di conseguenza le misure; è comunque sempre presente una componente non prevedibile, poiché tale disturbo può presentare delle fluttuazioni e dunque differire dal suo valore medio (flicker noise).

In ogni modo il fattore di scala e la componente costante del bias sono prevedibili e possono essere compensati tramite tarature iniziali e prove sperimentali, mentre l'instabilità del bias e il rumore bianco influenzano gli output in maniera non trascurabile, pertanto devono essere adottate delle strategie per poter ridurre la loro influenza sui risultati ottenuti attraverso i calcoli.

3 Modellizzazione e Simulazione della Camminata

3.1 Ricerca di modelli esistenti

Inizialmente è stata condotta una ricerca sui modelli matematici esistenti della camminata umana e si è notata subito l'estrema complessità dovuta alle numerose articolazioni coinvolte: il tentativo di riprodurre matematicamente tali movimenti sfocia in discipline come la biomeccanica e risulta particolarmente ostico tradurlo in un algoritmo. Successivamente è stato trovato un insieme di file con implementazioni in MATLAB di un modello di camminata che sfrutta un approccio dinamico, e distingue tra la fase di movimento della gamba (“swing”) e quella di riposo (“stance”). In questo modello si parte dall'idea di due aste (le gambe) congiunte all'altezza del bacino, con una massa M in corrispondenza della giunzione e due masse m in corrispondenza dei piedi; ci si concentra sull'andamento di alcuni angoli definiti in base al modello adottato, e tramite la variazione di momento angolare e la sommatoria dei momenti applicati, si arriva a calcolare l'accelerazione angolare, da cui attraverso successive integrazioni, all'andamento degli angoli. Da questi si ricava poi le coordinate dei piedi e della giunzione delle gambe nel tempo [4].

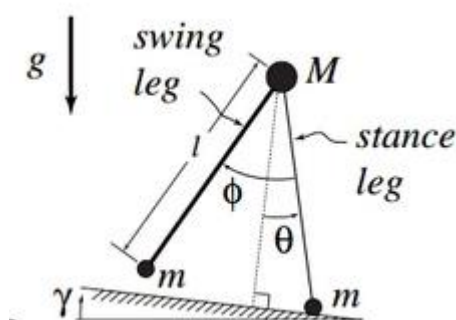


Figura 7: Modello dinamico di camminata

All'inizio è stato studiato questo modello e opportunamente modificato per calcolare velocità ed accelerazione dei piedi, per poter simulare l'acquisizione di dati da sensori inerziali. Ci si è accorti però della dubbia arbitrarietà dei parametri fisici utilizzati per permettere il funzionamento del modello, motivo per cui si è deciso realizzare un altro modello semplificato, partendo però da una schematizzazione simile.

3.2 Progetto di un modello cinematico

La soluzione che è stata adottata prevede sempre l'idea di due aste (le gambe) congiunte all'altezza del bacino, con i piedi all'altra estremità; si sono poi definiti opportunamente degli angoli di cui si è immaginato un andamento sinusoidale per simulare la cinematica della camminata in due dimensioni (figura 8).

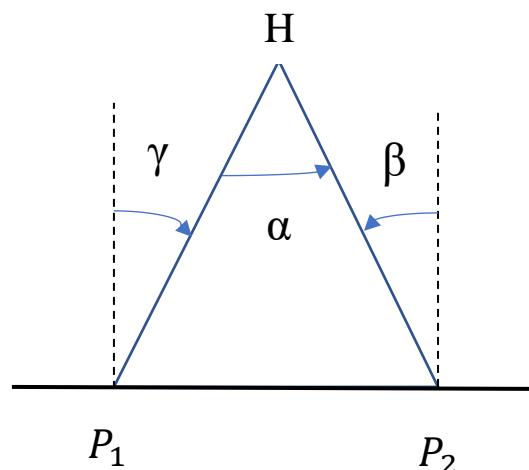


Figura 8: Modello cinematico della camminata

La posizione iniziale prevede il piede destro (P1) dietro, e quello sinistro (P2) avanti, di conseguenza gli angoli all'inizio della camminata saranno:

$$\begin{cases} \alpha = \alpha_{max} \\ \gamma = \gamma_{max} = \frac{\alpha_{max}}{2} \\ \beta = \beta_{max} = \frac{\alpha_{max}}{2} \end{cases}$$

Si nota anche che vale in generale la relazione:

$$\alpha = \beta + \gamma \quad (3.1)$$

In questa posizione iniziale è facile definire la lunghezza del passo l_p come la distanza lungo il terreno tra P_1 e P_2 , mentre la lunghezza della gamba è l_g (entrambi sono parametri modificabili nell'algorithm in MATLAB). Da queste lunghezze si può ricavare il valore di α_{max} :

$$\alpha_{max} = 2 \sin^{-1} \left(\frac{l_p}{2l_g} \right) \quad (3.1)$$

Dopodiché è stato adottato un andamento tramite la funzione coseno per tutti e tre gli angoli. Per farlo è stato necessario inserire il valore della velocità v (che può essere modificato come parametro iniziale), da cui è stato calcolato il periodo del ciclo T , che in questo caso, per analogia con la funzione coseno, corrisponde a due passi, e di conseguenza la pulsazione ω :

$$T = \left(\frac{2l_p}{v} \right) \quad (3.2)$$

$$\omega = \frac{2\pi}{T} \quad (3.3)$$

Ad ogni passo, dunque ogni mezzo ciclo, viene preso come riferimento per calcolare le coordinate necessarie il piede che si trova in stato stazionario; le funzioni che esprimono per ogni ciclo (in questo caso il primo) l'andamento

dei tre angoli quindi saranno:

$$0 < t < \frac{T}{2} \quad \begin{cases} \alpha = \alpha_{max} \cos(\omega t) \\ \gamma = \gamma_{max} \cos(\omega t) \\ \beta = \alpha - \gamma \end{cases} \quad (3.4)$$

$$\frac{T}{2} < t < T \quad \begin{cases} \alpha = \alpha_{max} \cos(\omega t) \\ \beta = \beta_{max} \cos(\omega t) \\ \gamma = \beta - \alpha \end{cases} \quad (3.5)$$

Con l'ipotesi di avere $l_p = 0.6 \text{ m}$, $l_g = 1 \text{ m}$ e $v = 1 \text{ m/s}$, allora si ottiene il seguente andamento degli angoli relativo a due cicli consecutivi (si noti che i grafici degli angoli β e γ coincidono):

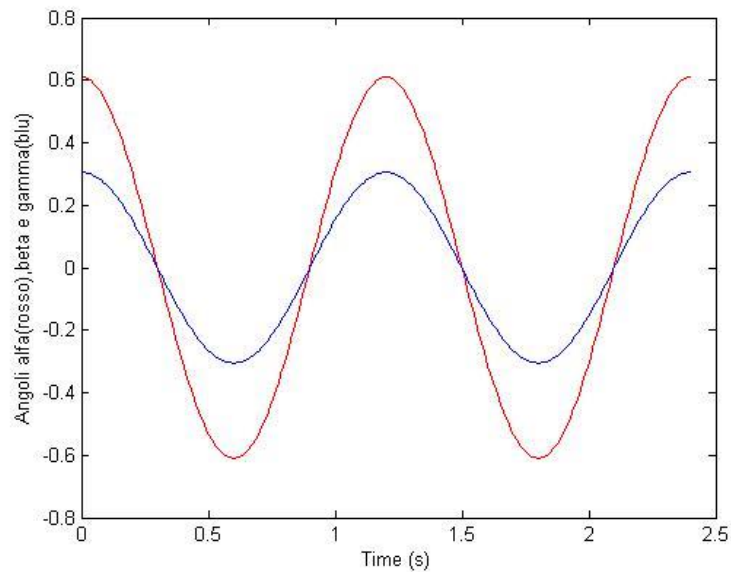


Figura 9: Andamento degli angoli per due cicli

Avendo poi sempre come riferimento il piede fermo (ad esempio nello stato iniziale è P_2), si possono calcolare la coordinate dell'altro piede e della congiunzione tra le gambe tramite le formule:

$$\begin{cases} x_H = x_2 - l_g \sin \beta \\ y_H = y_2 + l_g \cos \beta \end{cases} \quad (3.6)$$

$$\begin{cases} x_1 = x_H - l_g \sin \gamma = x_2 - l_g (\sin \beta + \sin \gamma) \\ y_1 = y_H + l_g \cos \gamma = y_2 + l_g (\cos \beta - \cos \gamma) \end{cases} \quad (3.7)$$

In base al numero di cicli desiderato, si tiene conto di quanti passi sono stati percorsi per stabilire quale piede risulta al momento stazionario e la sua posizione, e in base ad esso si calcolano le altre coordinate. Relazioni analoghe alla (3.6) e (3.7) si possono trovare nel caso in cui il piede P_1 sia quello stazionario. Il difetto principale di questo modello è l'assenza di simulazione di alzata del piede (come se strisciasse), per cui la coordinata y rimane nulla: è stato opportunamente modificato per compensare questo problema, facendo in modo che il piede descriva una circonferenza di raggio molto ampio, distaccandosi minimamente dal suolo.

3.3 Implementazione in MATLAB

Dopo aver ideato il modello cinematico lo si è sviluppato nel linguaggio MATLAB (Matrix Laboratory), un ambiente per il calcolo numerico e l'analisi statistica che consente di manipolare matrici e implementare algoritmi, molto comodo per simulazioni di questo tipo. Oltre all'interfaccia principale, per la scrittura e implementazione delle funzioni necessarie, è stato utilizzato il tool denominato Simulink, che permette la progettazione, la simulazione, la generazione automatica del codice e la verifica di sistemi embedded [5].

Inizialmente sono stati definiti i parametri che possono essere modificati e che risultano necessari per procedere nel resto dei calcoli:

- v : velocità di riferimento della camminata [m/s]
- l_p : lunghezza di un passo [m]
- l_g : lunghezza della gamba [m]
- n : numero di cicli da simulare (ogni ciclo corrisponde a due passi)
- f : frequenza di campionamento ipotetica (Sample Rate) [1/s]

Successivamente è stato realizzato l'algoritmo per simulare opportunamente il

modello matematico descritto nel paragrafo precedente, in modo da ottenere per ogni istante di campionamento le coordinate di entrambi i piedi, le coordinate della giunzione delle gambe (in due dimensioni) e il valore degli angoli di riferimento del modello. A partire dalle coordinate di un piede, è stato eseguito un rapporto incrementale per calcolare prima le velocità del piede, ed in seguito le accelerazioni (dt è il periodo di campionamento):

$$\begin{cases} v_x(t) \simeq \frac{x(t + dt) - x(t)}{dt} \\ v_y(t) \simeq \frac{y(t + dt) - y(t)}{dt} \end{cases} \quad (3.8)$$

$$\begin{cases} a_x(t) \simeq \frac{v_x(t + dt) - v_x(t)}{dt} \\ a_y(t) \simeq \frac{v_y(t + dt) - v_y(t)}{dt} \end{cases} \quad (3.9)$$

Lo scopo è quello di simulare l'ottenimento di dati da una piattaforma IMU, contenente due accelerometri e un giroscopio trasversale, situata nel piede, che segua dunque la sua inclinazione. All'interno di questo modello è stato supposto che il piede rimanga sempre perpendicolare alla gamba, da cui il sistema di riferimento assi body è dato dall'asse 1 lungo la direzione del piede ed uscente dalla parte anteriore di esso, l'asse 2 lungo la gamba e verso la congiunzione, mentre l'asse 3 per ottenere una terna destrorsa (figura 10).

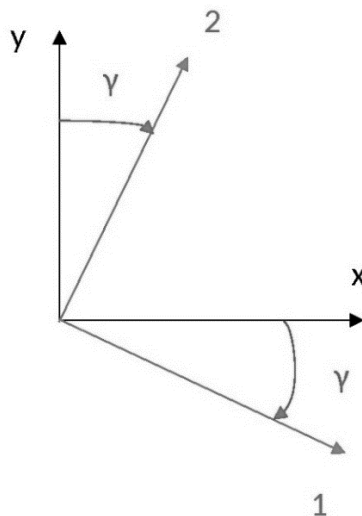


Figura 10: Sistemi di riferimento

A questo punto è stata applicata la matrice di rotazione dal sistema xy al sistema assi body:

$$\begin{pmatrix} a_1 \\ a_2 \end{pmatrix} = C_{xy}^{body} \begin{pmatrix} a_x \\ a_y \end{pmatrix} = \begin{pmatrix} \cos \gamma & -\sin \gamma \\ \sin \gamma & \cos \gamma \end{pmatrix} \begin{pmatrix} a_x \\ a_y \end{pmatrix} \quad (3.10)$$

Dopo aver ottenuto le accelerazioni in assi body secondo la (3.10), è stato simulato l'ottenimento di dati affetti da un disturbo come rumore bianco con la somma di un valore random per ogni istante (essendo infatti una sequenza di valori del tutto casuali a frequenza più elevata rispetto al campionamento del segnale). Il valore della deviazione standard del rumore simulato può essere modificato come parametro.

Inserendo i parametri con i seguenti valori:

- $v = 1$ [m/s]
- $l_p = 0.6$ [m]
- $l_g = 1$ [m]
- $n = 2$
- $f = 100$ [1/s]

Si ottiene come mostrato in figura 11 l'andamento delle accelerazioni in assi body misurate da un sensore e affette da un rumore bianco con deviazione standard pari a tre (si ipotizza che gli altri tipi di errori siano stati già compensati).

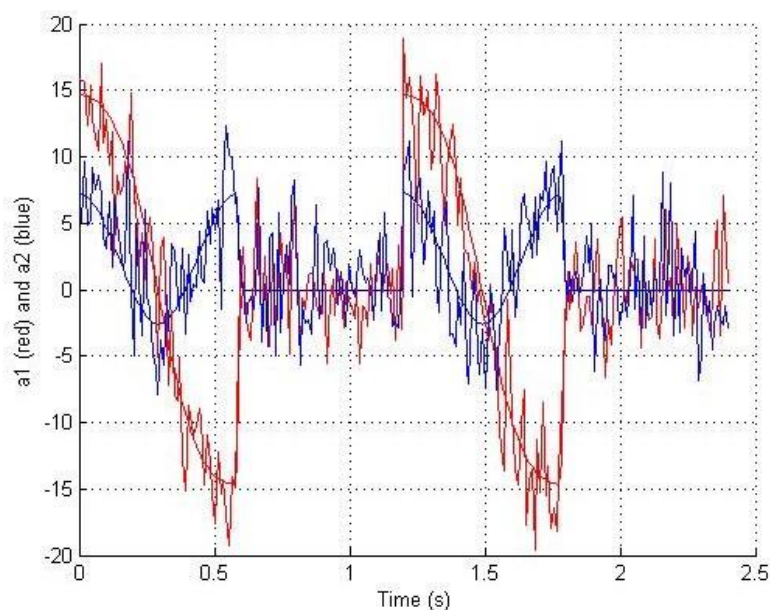


Figura 11: Accelerazioni misurate in assi body

3.4 Progettazione del filtro

Successivamente è stato progettato un filtro “passa-basso” per poter eliminare quelle frequenze legate al disturbo, ripulendo il più possibile il segnale. Lo studio di questa parte è stato condotto ragionando nel dominio complesso o dominio s , in relazione al concetto di “Trasformata di Laplace”, che, ad una funzione $f(t)$ nel dominio del tempo, associa una funzione $F(s)$ nel dominio complesso:

$$L\{f(t)\} = \int_0^{+\infty} f(t)e^{-st} dt =: F(s) \quad (3.11)$$

In questo caso facciamo riferimento a due funzioni: quella di ingresso è data dall’andamento delle accelerazioni come fossero misurate dal sensore, perciò affette dal disturbo, mentre quella di uscita rappresenta il segnale filtrato. Nel caso di ingresso nullo, anche l’uscita sarà nulla, perciò la risposta libera in questo sistema risulta nulla. Chiamando $Y(s)$ la trasformata dell’uscita e $X(s)$ la trasformata dell’ingresso, la funzione di trasferimento sarà:

$$G(s) = \frac{Y(s)}{X(s)} \quad (3.12)$$

Per realizzare il filtro, la funzione di trasferimento è stata pensata come sistema del primo ordine senza zeri, e con costante di guadagno $K' = 1$. Nella forma in cui viene evidenziata la costante di tempo τ si ottiene [7]:

$$G(s) = \frac{1}{1 + \tau s} \quad (3.13)$$

La costante di tempo è indicativa di quali frequenze del segnale vengano lasciate passare e quali vengano filtrate: più la costante di tempo è elevata, più il segnale viene ripulito dalle frequenze più elevate. Per trovare il valore più adatto al nostro caso, è stato sfruttato il tool di MATLAB Simulink (figura 12), ricreando il segnale disturbato e applicando la funzione di trasferimento

indicata in (3.13). Si può notare che, utilizzando valori di τ intorno ad 1 o 2, il segnale ottenuto è ripulito in maniera ottimale, ma risulta traslato nel tempo, in particolare in ritardo. Occorre dunque trovare un compromesso per ottenere un segnale abbastanza ripulito ma non troppo in ritardo: dopo diverse prove effettuate anche analizzando i risultati nel simulatore della camminata, per i parametri inseriti, si è arrivati ad un valore ottimale di τ intorno a 0.03.

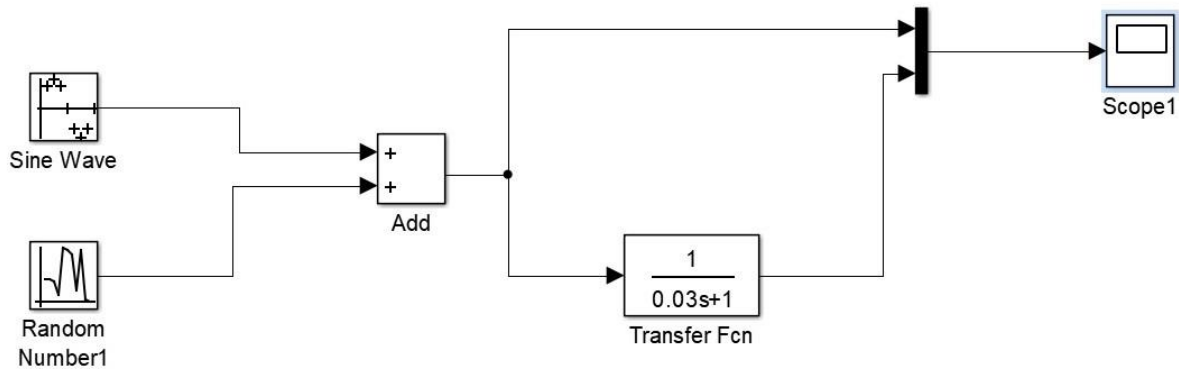


Figura 12: Calibrazione di τ in Simulink

Partendo dalla (3.12) e dalla (3.13), occorre trovare un'espressione da poter implementare in MATLAB esprimendo la funzione di uscita in relazione al segnale in ingresso delle accelerazioni:

$$\frac{1}{1 + \tau s} = \frac{Y(s)}{X(s)} \quad (3.14)$$

$$(1 + \tau s)Y(s) = X(s)$$

$$Y(s) + \tau s Y(s) = X(s)$$

Tornando al dominio dei tempi, ed in particolare tenendo conto del teorema della derivata generalizzata per le trasformate di Laplace

$$L\{\dot{y}(t)\} = sY(s) - y(0^-)$$

Si ottiene

$$y(t) + \tau \dot{y}(t) = x(t)$$

$$\dot{y}(t) = -\frac{1}{\tau}y(t) + \frac{1}{\tau}x(t)$$

Approssimando poi la derivata col rapporto incrementale si ottiene

$$\frac{y(t + dt) - y(t)}{dt} \simeq -\frac{1}{\tau}y(t) + \frac{1}{\tau}x(t)$$

Con qualche passaggio si arriva alla forma finale:

$$y(t + dt) = y(t) \left(1 - \frac{dt}{\tau}\right) + \frac{dt}{\tau} x(t) \quad (3.15)$$

Ponendo la condizione iniziale che $y(0) = x(0)$, si possono dunque trovare tutti i valori dell'accelerazione con l'applicazione del filtro progettato (figura 12).

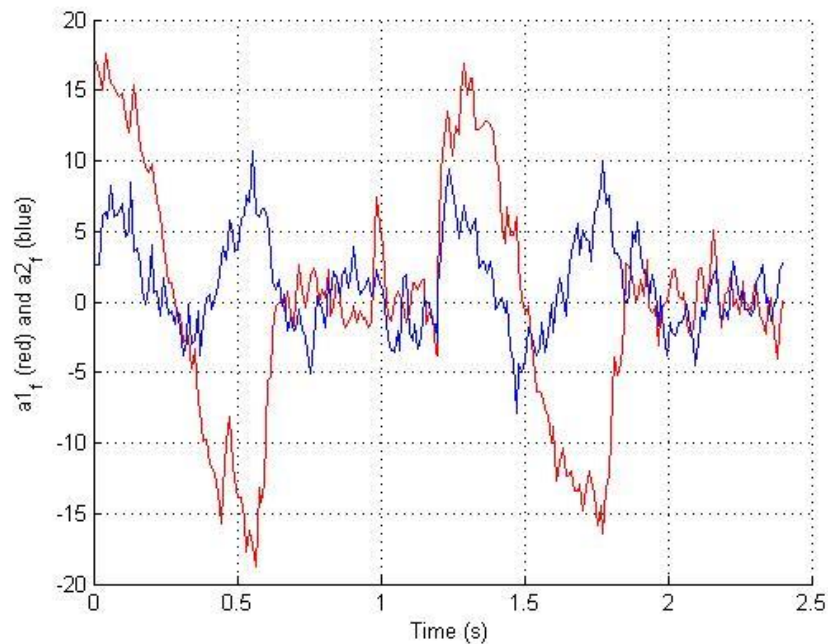


Figura 13: Accelerazioni ottenute con l'applicazione del filtro

3.5 Calcolo dell'assetto (angolo γ)

Nella “pedestrian navigation” la piattaforma IMU è inserita nella suola della scarpa, dunque gli accelerometri seguono il sistema di riferimento del piede (assi body), che è differente da quello della navigazione. Per poter sfruttare le accelerazioni e procedere con l'integrazione per ottenere la posizione, è prima necessario avere a disposizione gli angoli e la matrice di rotazione ad essi legata per trasformare le accelerazioni nel sistema di navigazione.

Nel simulatore progettato, l'angolo γ è quello che ci permette di passare da un sistema di riferimento all'altro, come è evidente nella figura 10. Questo angolo può essere ottenuto da un giroscopio lungo l'asse 3 del sistema body nel modello ideato, e per simulare l'acquisizione di dati da un sensore simile è stata calcolata la derivata dell'andamento dell'angolo γ per ogni istante, che coincide con la velocità angolare:

$$\dot{\gamma}(t) \simeq \frac{\gamma(t + dt) - \gamma(t)}{dt} \quad (3.16)$$

Analogamente al caso dell'accelerazione, per simulare l'acquisizione di dati reali è stato aggiunto un disturbo (un rumore bianco) alle velocità angolari, con deviazione standard inserita come parametro. A questo punto è stato applicato un filtro per ripulire il segnale, secondo il procedimento illustrato al paragrafo precedente, e valutando il valore ottimale della costante di tempo con un modello in Simulink simile a quello in figura 12.

Prima di poter procedere con l'integrazione della velocità angolare e ricavare l'andamento dell'angolo γ , è necessario calcolare il valore iniziale di quest'ultimo tramite i dati a disposizione. Con l'ipotesi che nel primo istante di campionamento il piede sia fermo, sono stati sfruttati i dati degli accelerometri, utilizzati dunque come inclinometri, secondo il principio che porta alla relazione (2.6) calcolata precedentemente:

$$\gamma_0 = \sin^{-1} \left(\frac{a_1}{g} \right) \quad (3.17)$$

Di conseguenza è stato calcolato un ipotetico andamento dell'angolo γ :

$$\gamma(t + dt) = \dot{\gamma}_f(t)dt + \gamma(t) \quad (3.18)$$

Dove $\dot{\gamma}_f$ è il valore della velocità angolare ottenuto a seguito dell'applicazione del filtro. In figura 14, il rosso indica l'andamento della velocità angolare reale, il giallo quello con l'aggiunta dei disturbi, ed il blu dopo l'applicazione del filtro (con costante di tempo 0.04).

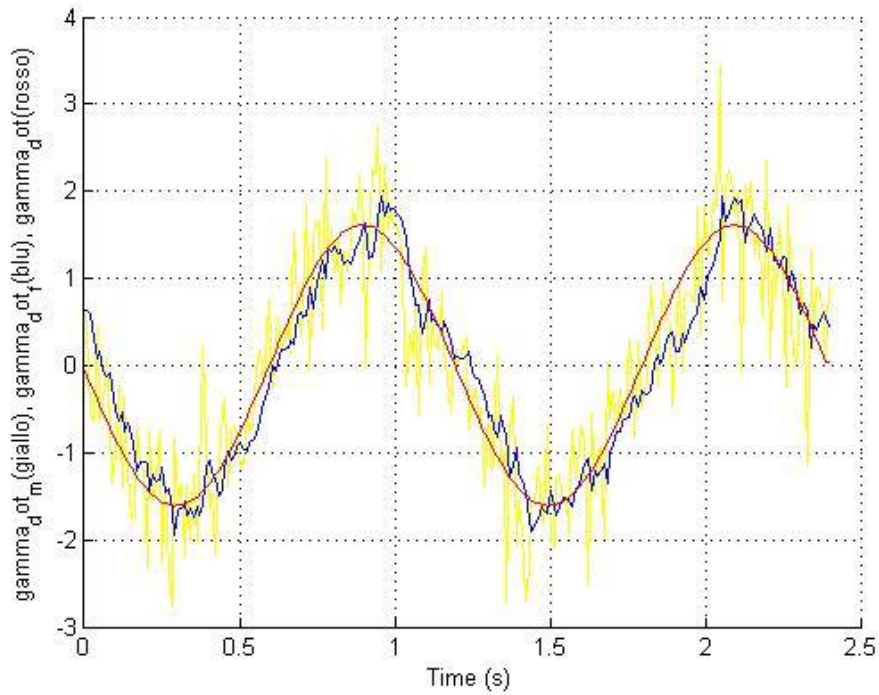


Figura 14: Andamento dei diversi $\dot{\gamma}$

4 Algoritmi

In questa sezione verranno descritti i vari algoritmi che sono stati o progettati, oppure analizzati e opportunamente modificati per lo studio di alcune caratteristiche della camminata. Inizialmente verrà esposta la fase di integrazione in relazione alla simulazione della camminata del paragrafo precedente, per poi passare ad un primo tentativo sperimentale tramite un'applicazione già esistente per gli smartphone ed infine descrivere dal punto di vista teorico il programma del progetto "OpenShoe" già citato.

4.1 Integrazione

Avendo a disposizione le misure delle accelerazioni e dell'angolo ripulite dai disturbi, è possibile procedere con l'integrazione. Innanzitutto, si possono calcolare le accelerazioni nel sistema di riferimento di navigazione, con una relazione simile alla (3.10):

$$\begin{pmatrix} a_x \\ a_y \end{pmatrix} = C_{body}^{xy} \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} \cos \gamma & \sin \gamma \\ -\sin \gamma & \cos \gamma \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \quad (4.1)$$

Tramite l'accelerazione a_x si ottiene con un'integrazione approssimata l'andamento della velocità v_x e della posizione x .

$$v_x(t + dt) \simeq a_x(t)dt + v_x(t) \quad (4.2)$$

$$x(t + dt) \simeq v_x(t)dt + x(t) \quad (4.3)$$

La posizione finale ottenuta può essere confrontata con quella reale che deriva dal moltiplicare il numero dei cicli per il doppio della lunghezza del passo, per poi calcolare una stima dell'errore relativo. Attraverso le sole relazioni

(4.2) e (4.3) sono stati ottenuti errori molto elevati, e con un'analisi dell'andamento della velocità è stato notato che anche negli intervalli di tempo in cui ci si aspettava un valore nullo (piede stazionario), in realtà non lo era. Ciò è spiegabile con il disturbo aggiunto per simulare l'acquisizione di un dato reale, perciò si è pensato di inserire il concetto di “Zero-Velocity Update” (ZUPT) [8]. Questo è legato alla periodicità della camminata umana, per cui ad ogni ciclo il piede risulta stazionario (“stance phase”) in un certo intervallo di tempo ed è possibile in questi istanti imporre che la velocità sia nulla.

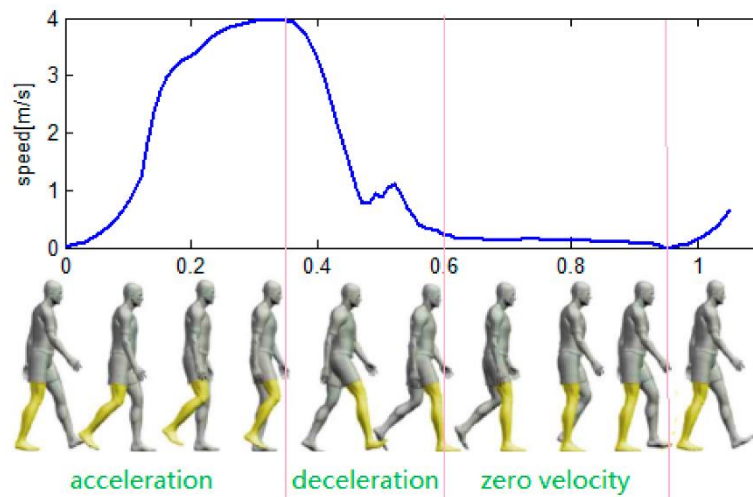


Figura 15: Ciclo della camminata

Per rilevare gli istanti di tempo in cui il piede si trova nella “stance phase” ci sono diverse possibilità e nel caso di questo simulatore è stato utilizzato il modulo dell'accelerazione totale: ogniqualvolta il modulo risulta al di sotto di una certa soglia, viene imposta la velocità a zero. In questo modo l'integrazione viene reinizializzata periodicamente, e si evita di continuare ad accumulare l'errore dovuto ai disturbi. Indicando il modulo dell'accelerazione $A(t)$

$$A(t) = \sqrt{a_x^2 + a_y^2} = \sqrt{a_1^2 + a_2^2}$$

si può completare la relazione (4.2) in questo modo (s è la soglia del modulo al di sotto della quale si impone velocità nulla):

$$v_x(t) = 0 \quad \text{per } A(t) < s \quad (4.4)$$

Con questo accorgimento, si ottengono grafici della velocità e posizione di un piede come mostrato in figura 16 e 17 (caso con $n=2$), ed errori relativi di posizione finale che in media sono intorno al 5-10%.

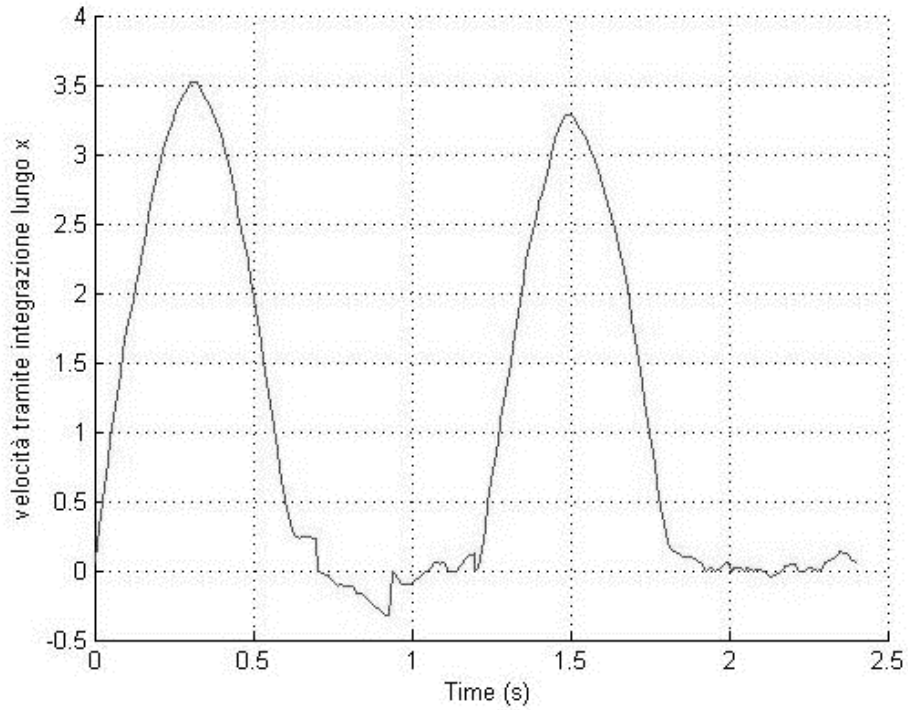


Figura 16: Velocità lungo x tramite integrazione

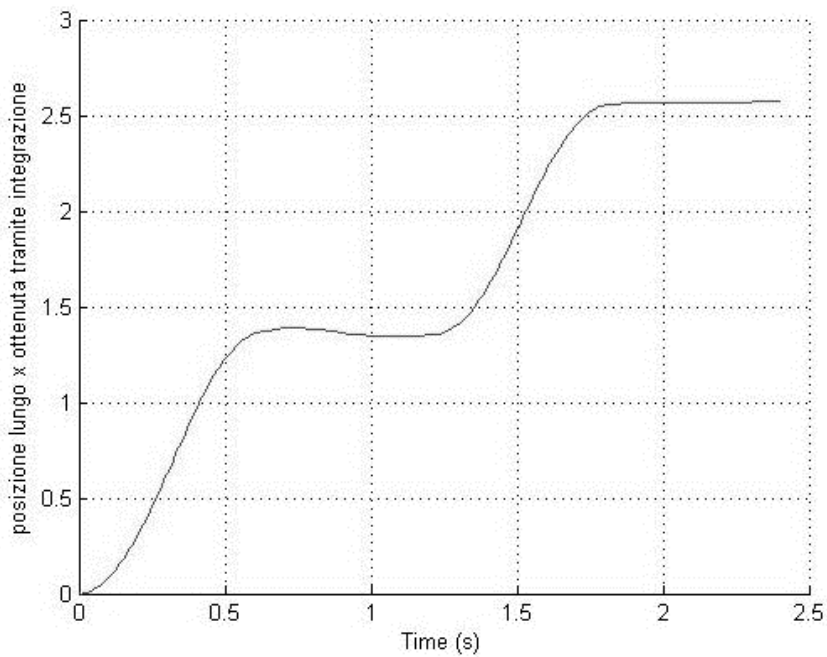


Figura 17: Posizione x tramite integrazione

4.2 Applicazione “Sensor Kinetics”

Il passo successivo è stato quello di provare ad ottenere dati sperimentali partendo dai sensori presenti negli smartphone, per poi elaborarli secondo l’algoritmo presentato sopra e stimare la distanza percorsa. È stata individuata l’applicazione Sensor Kinetics, disponibile sia per Android che per iPhone, realizzata da RotoView [9]. Essa permette di monitorare in tempo reale le misure dei sensori disponibili e attivi sul proprio smartphone tra accelerometri, giroscopi, magnetometri. A partire da questi dati, sono già implementati degli algoritmi per simulare la presenza di sensori d’assetto, sensori di accelerazione lineare e sensori di gravità. L’applicazione è molto semplice da utilizzare, e contiene diversi moduli che aiutano a sfruttarla e spiegano come realizzare esperimenti per capire meglio il funzionamento dei sensori MEMS e di alcuni principi fisici. La versione “pro” (a pagamento) consente inoltre di registrare le misurazioni in intervalli di tempo desiderati, per poi riceverle per mail tabulati in un formato predefinito: ciò è risultato utile per inserire i dati nell’algoritmo sviluppato e provare a calcolare la distanza percorsa.



Figura 18: Schermata principale dell'applicazione Sensor Kinetics

Il cellulare utilizzato è il modello LG G2 mini, che però non contiene attivati i giroscopi, motivo per cui sono visualizzabili solamente le misure di accelerometri, magnetometri, e sensori di assetto: questi ultimi calcolano i valori a partire dalle misure degli accelerometri stessi.

L'applicazione è stata utilizzata durante una camminata di otto passi in un terreno in piano, legando il cellulare al piede e registrando i dati dagli accelerometri e dai sensori di assetto e classificandoli in un formato utilizzabile in MATLAB. I grafici ottenuti sono i seguenti:



Figura 19: Grafici delle accelerazioni (sinistra) e degli angoli di assetto durante gli 8 passi

Trasformando le accelerazioni con un'opportuna matrice di rotazione nel sistema di navigazione e procedendo con l'integrazione secondo quanto esposto nel paragrafo precedente, è stata ricavata una misura della distanza percorsa, che però risulta fortemente errata, addirittura a volte quasi nulla o negativa. Le spiegazioni principali per questi errori sono i disturbi e le derive che comportano la necessità di un'elaborazione dei dati particolare, nonché il calcolo degli angoli molto impreciso, a causa della mancanza dei giroscopi (il segnale

relativo all'angolo di assetto risulta ritardato se calcolato con i soli accelerometri).

4.3 Contapassi

Data l'inutilizzabilità delle misure ottenute tramite l'applicazione per misurare la distanza percorsa direttamente, e notando la periodicità dei grafici in figura 19, è stato pensato di realizzare un algoritmo che contasse il numero di passi effettuati, di modo che sia possibile arrivare ad una stima della distanza impostando una certa lunghezza del passo.

L'idea di base è quella di prendere le misure ottenute dai tre accelerometri presenti nel cellulare, e calcolarne il modulo istante per istante. Dopodiché si procede individuando i massimi relativi selezionando i diversi intervalli di campionamento entro un limite opportuno e si valuta se questi massimi relativi del modulo superano o meno una certa soglia che deve essere impostata. Ogniqualevolta il massimo relativo supera la soglia, un contatore incrementa il proprio valore di uno.

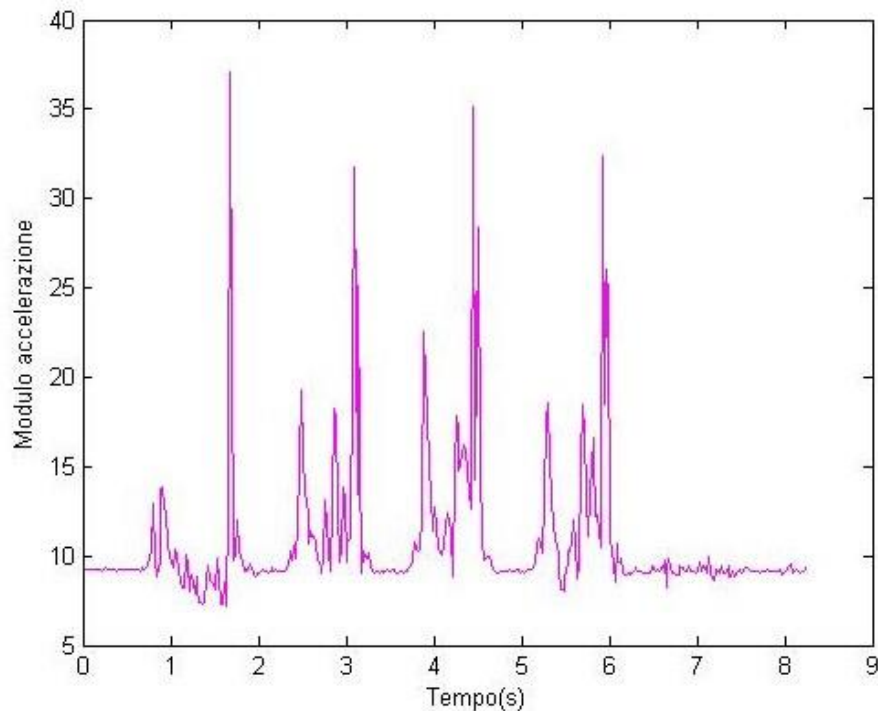


Figura 20: Andamento del modulo dell'accelerazione

In generale l'andamento del modulo dell'accelerazione presenta ciclicamente due picchi particolarmente evidenti: quello relativo all'impatto del piede col terreno, e quello causato dall'alzata improvvisa del piede (di intensità minore). In questo caso è stata impostata una soglia ragionevolmente alta per far sì che il contatore incrementi solo con l'impatto del piede sul terreno, e successivamente venga raddoppiato il suo valore per ricavare il numero di passi effettivo. Questo sistema presenta l'errore di un passo in più o in meno, nel caso in cui il primo piede a muoversi non sia quello che viene monitorato dal sensore ma l'altro. Nel nostro caso, l'algoritmo rileva con successo gli 8 passi effettuati.

4.4 Progetto OpenShoe

Si è visto precedentemente che il tentativo di integrazione a partire dai dati misurati dai sensori inerziali del cellulare non ha portato a risultati soddisfacenti: oltre a difetti dei sensori stessi (solitamente più economici negli smartphone), ciò è legato anche all'algoritmo utilizzato, poiché è necessaria un'opportuna elaborazione dei segnali in input per compensare la presenza di tutti quegli errori e disturbi elencati nel capitolo 2. Dopo diverse ricerche, è stato individuato il progetto OpenShoe, organizzato e realizzato presso il KTH di Stoccolma nel laboratorio di elaborazione dei segnali. I prossimi paragrafi riguarderanno lo studio e l'analisi dei codici disponibili open-source, nonché la trattazione teorica che ne è la base, per capire quali sono le modalità per rendere efficace l'utilizzo dei dati forniti dai sensori inerziali e ottenere un risultato soddisfacente nell'ambito della "pedestrian navigation".

Il progetto mette a disposizione descrizioni dei componenti hardware utilizzati e software elaborati per realizzare un sistema in grado di calcolare la posizione raggiunta dal piede rispetto al punto di partenza, attraverso un complesso procedimento di integrazione dei dati ottenuti dall'IMU, con filtro di Kalman e "Zero-Velocity Update" (ZUPT) [10]. La parte hardware comprende proprio la piattaforma IMU collegata ad un microprocessore, il tutto inserito all'interno di un contenitore da posizionare nella suola della scarpa, la posizione più adatta per ottenere buone misurazioni, data la vicinanza col terreno (figura 21). Nel lavoro di tesi sono stati utilizzati e analizzati i file per le implementazioni MATLAB ed anche le cartelle contenenti i dati ricavati dalle misurazioni della piattaforma inerziale, una MicroStrain 3DX-GX2 IMU con

range dinamico di $\pm 18g$ per gli accelerometri e di 1200 deg/s per i giroscopi, ed una frequenza di campionamento pari a 250 Hz [11]. I dati vengono memorizzati per l'elaborazione in post-processing.



Figura 21: Piattaforma IMU del progetto OpenShoe

In generale, il sistema può essere descritto seguendo lo schema in figura 22: i dati vengono ottenuti dall'IMU ed elaborati opportunamente in modo da ottenere l'accelerazione nel sistema di navigazione, da cui viene poi sottratta facilmente la componente gravitazionale. Dall'accelerazione vettoriale lineare si può calcolare così attraverso l'integrazione la velocità e la posizione; l'assetto viene ottenuto a partire dalle misure dei giroscopi ed effettuando l'integrazione. È già stato sottolineato il problema secondo cui, un sistema di questo tipo, continua ad accumulare errori dovuti ai difetti dei sensori MEMS e conduce così ad un risultato della posizione inutilizzabile già dopo una decina di secondi [11].

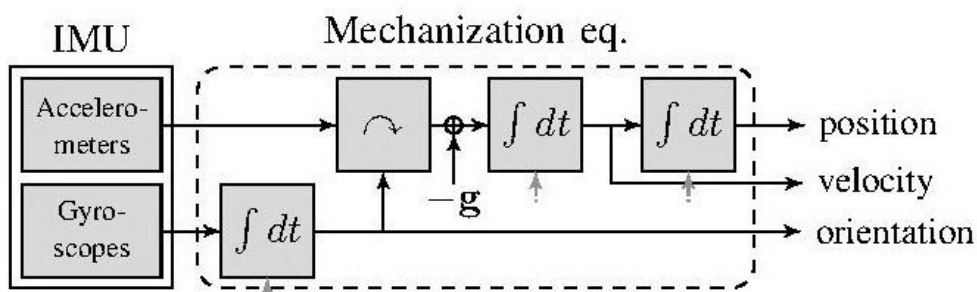


Figura 22: Schema del funzionamento del sistema inerziale e delle integrazioni effettuate

Per risolvere questa situazione, si sfrutta la stazionarietà del piede che avviene ciclicamente ogniqualvolta dopo l’impatto col terreno esso è sottoposto ad accelerazioni e velocità di rotazione praticamente nulle. Queste condizioni si verificano però contemporaneamente solo durante alcuni istanti, ed è necessario regolare il rilevatore (ZUPT detector) in maniera da discriminare lo stato stazionario efficacemente. Nel caso in cui sia rilevato lo stato stazionario, si sfrutta il filtro di Kalman (figura 23) che stima l’errore che si è accumulato nel tempo sulla base della discrepanza tra la velocità calcolata e l’ipotesi secondo cui dovrebbe essere nulla, per correggere così lo stato nel sistema di navigazione inerziale (posizione, velocità e assetto). Esistono diverse modalità per calibrare il rilevatore di stato stazionario del piede, e verranno descritte in seguito per poi valutare quale sia la più performante. Nei paragrafi successivi verrà descritto il codice MATLAB fornito, nonché spiegati i passaggi teorici che ne sono la base.

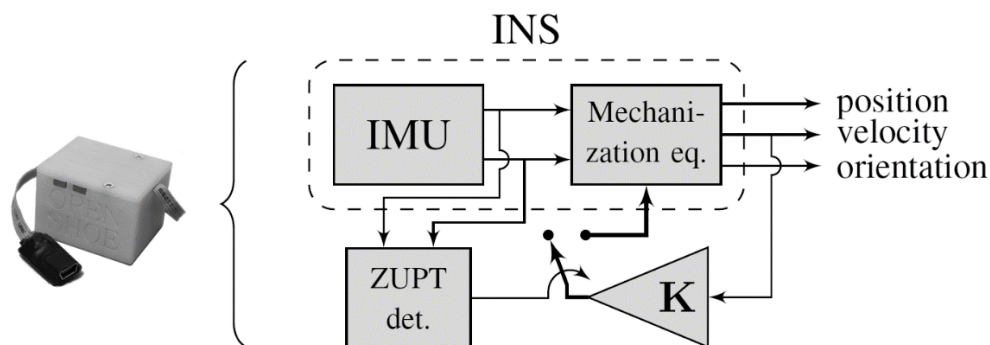


Figura 23: Schema dell'INS integrato con ZUPT detector

4.5 Impostazioni iniziali

Nella parte iniziale del programma (“settings”) vengono definiti dei parametri necessari per i calcoli successivi, e vengono caricati i dati IMU che sono stati raccolti in un file. Per raccogliere insieme tutti questi parametri inizializzati, viene utilizzata una variabile globale (“global” in MATLAB) *simdata*, in modo da poterne condividere i valori con tutte le funzioni del programma.

Vengono inseriti i valori dell'altitudine h [m] e della latitudine Φ [gradi sessagesimali] di riferimento della località in cui vengono effettuate le misurazioni, che poi verranno utilizzati per calcolare il valore dell'accelerazione gravitazionale g secondo il modello WGS84 (World Geodetic System 1984), sistema di coordinate basato sulla Terra modellata come ellissoide di rotazione, con piano di simmetria verticale (figura 24). In questo caso la latitudine geodetica è l'angolo tra il piano equatoriale e la verticale locale, ovvero la retta che passa per il punto di riferimento e la sua proiezione sull'ellissoide, tale inoltre da essere ortogonale al piano tangente alla superficie terrestre passante per la proiezione stessa; l'altitudine è la distanza tra il punto e la sua proiezione sull'ellissoide lungo la verticale [12].

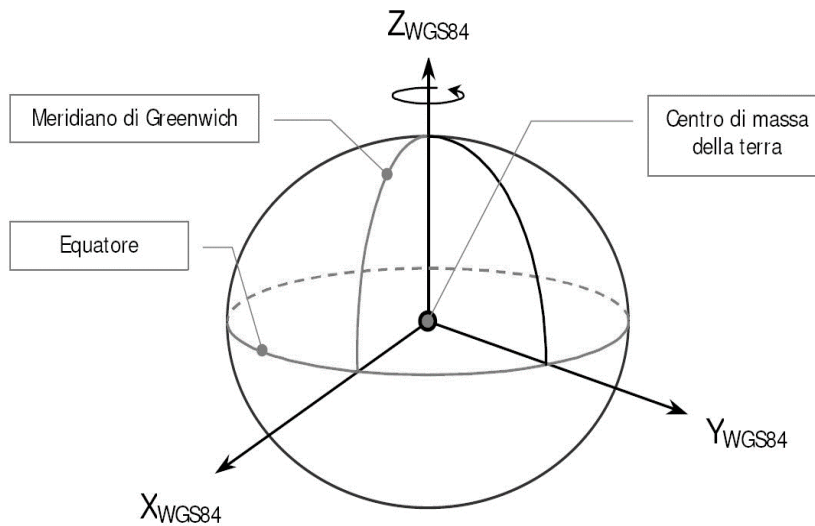


Figura 24: Modello della Terra WGS84

Attraverso le funzioni MATLAB `fopen`, `fscanf`, `fclose` viene caricato il file contenente i dati dalla piattaforma IMU, raccolti in una matrice che ha come righe le tre accelerazioni e le tre velocità angolari, mentre come colonne i rispettivi valori ad ogni istante di campionamento. I valori grezzi di accelerazione ottenuti dall'IMU vengono moltiplicati per un fattore di scala indicato dalla stessa casa produttrice, pari a 9.80665.

La frequenza di campionamento viene impostata secondo le caratteristiche della piattaforma a 250 Hz; viene poi inserito il vettore della posizione iniziale ed anche l'angolo di direzione (heading) iniziale, mentre i restanti angoli di assetto (roll e pitch) vengono calcolati a partire dalla media dei primi venti valori ottenuti dagli accelerometri, utilizzati dunque come inclinometri.

Per quanto riguarda il ZUPT detector, è possibile scegliere la metodologia matematica con cui rilevare lo stato stazionario e la soglia che permette di distinguerlo; inoltre è possibile inserire i valori delle deviazioni standard dei rumori di accelerometri e giroscopi.

Gli altri parametri inseriti riguardano il filtro per la stima dello stato (rumore di processo, rumore di uscita, e valori delle varianze da inserire nella matrice di covarianza dell'errore dello stato), pertanto è necessario introdurre i concetti teorici ad essi legati.

4.6 Modello ISU del Sistema e Osservatore dello Stato

In campo ingegneristico per la descrizione dei sistemi viene spesso utilizzata la schematizzazione tramite modello ISU (Ingresso-Stato-Uscita). Denominando con \bar{x} il vettore che rappresenta lo stato del sistema, ovvero l'insieme delle grandezze utili per capire come si sta comportando, con \bar{u} il vettore degli ingressi e con \bar{y} il vettore delle uscite, si rappresenta il modello in questo modo:

$$\begin{cases} \dot{\bar{x}} = f(\bar{x}, \bar{u}, t) \\ \bar{y} = h(\bar{x}, \bar{u}, t) \end{cases} \quad (4.5)$$

La funzione f è l'equazione di stato (modello del processo), mentre la h il modello delle uscite. Supponendo che il vettore di stato sia di n componenti, quello degli ingressi di r componenti, e quello delle uscite di m , allora la prima non è altro che un sistema di n equazioni differenziali (risulta perciò necessario avere le n condizioni iniziali $\bar{x}(0) = \bar{x}_0$), mentre la seconda un sistema di m equazioni algebriche. Questo è il caso più generale in cui il sistema sia non lineare e tempo continuo [13].

Per lo studio dell'evoluzione dello stato nella pedestrian navigation, viene invece introdotto il modello ISU tempo-discreto, dove è opportuno definire i vari istanti di campionamento t_k (k numero intero) in relazione al passo di campionamento T (sample time), inverso della frequenza di campionamento:

$$t_k = t_{k-1} + T = t_0 + kT \quad (4.6)$$

Nei modelli tempo discreti infatti l'operatore derivata temporale non è impiegabile poiché non è possibile definire il limite del periodo temporale che tende a zero, dunque si fa uso delle equazioni alle differenze, esprimendo il valore di una variabile in un certo istante in funzione del valore negli istanti precedenti. Supponendo inoltre che il sistema sia lineare, si ricava il seguente modello di riferimento:

$$\begin{cases} \bar{x}_{k+1} = A_d \bar{x}_k + B_d \bar{u}_k \\ \bar{y}_k = C_d \bar{x}_k + D_k \bar{u}_k \end{cases} \quad (4.7)$$

A_d : matrice di stato (di transizione)

B_d : matrice di distribuzione degli ingressi

C_d : matrice delle uscite

D_k : matrice di collegamento algebrico input/output

Adattando la (4.7) al caso trattato da questa tesi, lo stato è un vettore di 9 elementi (errori di posizione, velocità, assetto), l'ingresso un vettore di 6 (dati da accelerometri e giroscopi) e l'uscita è ciò che ci interessa monitorare dello stato, in particolare l'errore della velocità. Aggiungendo il rumore di processo \bar{w}_k , che condiziona l'evoluzione dello stato, ed il rumore di uscita \bar{v}_k , si arriva alla seguente forma dell'evoluzione degli errori indicati nel vettore di stato:

$$\begin{cases} \bar{x}_{k+1} = F \bar{x}_k + G \bar{w}_k \\ \bar{y}_k = H \bar{x}_k + \bar{v}_k \end{cases} \quad (4.8)$$

in cui G è la matrice che moltiplica il rumore del processo, mentre H è la matrice di osservazione dello stato (in questo caso B_d è una matrice nulla). I due rumori introdotti sono segnali determinati dai sensori utilizzati, e vale l'ipotesi che siano rumori bianchi, ovvero a media nulla nel tempo:

$$E[\bar{w}_k] = E[\bar{v}_k] = 0 \quad (4.9)$$

Per quanto riguarda le loro matrici di covarianza, si ipotizza che siano diagonali, in cui dunque ci siano lungo la diagonale principale i valori delle varianze degli elementi dei vettori, e zero nel resto della matrice. Indicando con

I_3 e 0_3 la matrice identità e quella nulla 3×3 , e con σ_a^2 e σ_ω^2 le varianze sulle accelerazioni e velocità di rotazione rispettivamente, si possono definire:

$$Q \triangleq E[\bar{w}_k \bar{w}_k^T] = \begin{bmatrix} \sigma_a^2 I_3 & 0_3 \\ 0_3 & \sigma_\omega^2 I_3 \end{bmatrix} \quad (4.10)$$

$$R \triangleq E[\bar{v}_k \bar{v}_k^T] = \sigma_v^2 I_3 \quad (4.11)$$

Q : matrice di covarianza del rumore di processo

R : matrice di covarianza della pseudo-misura del rumore (dell'errore di misura della velocità)

Valgono inoltre le seguenti ipotesi di cross-correlazione e di correlazione nel tempo:

$$E[\bar{w}_k \bar{v}_k^T] = 0 \quad (4.12)$$

$$E[\bar{w}_k \bar{w}_j^T] = 0 \quad (4.13)$$

$$E[\bar{v}_k \bar{v}_j^T] = 0 \quad (4.14)$$

Le formule (4.12-14) esprimono che i due tipi di errori sono indipendenti tra di loro e inoltre lo stesso errore in un istante risulta non correlato rispetto all'istante precedente.

Il passo successivo è quello di inserire il modello dell'osservatore dello stato (nell'ipotesi che sia osservabile), che permette di ottenere una stima dello stato \hat{x} nel tempo, e verrà poi sfruttato per correggere l'integrazione del sistema inerziale (figura 25). Nella notazione successiva, si indicherà con apice $-$ e $+$ rispettivamente prima e dopo aver incorporato le misure al modello [14].

$$\begin{cases} \hat{x}_{k+1}^- = F\hat{x}_k^+ \\ \hat{y}_k^- = H\hat{x}_k^- \end{cases} \quad (4.15)$$

Nei passaggi successivi per evitare ridondanza nella notazione si ometterà il simbolo di vettore.

Nel momento in cui la misura y_k diventa disponibile, si può calcolare l'innovazione i_k (che corregge il modello dell'osservatore):

$$i_k = y_k - \hat{y}_k^- \quad (4.16)$$

Introducendo la matrice dei guadagni dell'osservatore K_k , si può ricavare la stima dello stato corretta:

$$\hat{x}_k^+ = \hat{x}_k^- + K_k i_k \quad (4.17)$$

Definendo inoltre il vettore dell'errore dello stato

$$\begin{cases} e_k^- = x_k - \hat{x}_k^- \\ e_k^+ = x_k - \hat{x}_k^+ \end{cases} \quad (4.18)$$

e introducendo le seguenti relazioni dimostrabili [14]:

$$\begin{cases} e_{k+1}^- = F e_k^+ + w_k \\ i_k^- = H e_k^- + v_k \end{cases} \quad (4.19)$$

Si arriva alla forma per esprimere come varia l'errore prima e dopo aver integrato le misurazioni:

$$e_k^+ = (I - K_k H) e_k^- + K_k v_k \quad (4.20)$$

In relazione al nostro caso, la (4.20) esprime l'entità della correzione che dovrà essere applicata allo stato stimato quando verrà rilevato lo stato stazionario.

Avendo introdotto l'errore di stima e_k , si possono introdurre le seguenti ipotesi:

$$\begin{cases} E[e_k] = 0 \\ E[e_k e_k^T] = P_k \end{cases} \quad (4.21)$$

La prima relazione indica che la misura è accurata, ovvero l'errore di stima in media nel tempo risulta nulla, mentre la seconda definisce la matrice di covarianza dell'errore di stima dello stato. Analogamente alle matrici Q, R già introdotte, anche la P si ipotizza essere una matrice diagonale.

Per quanto riguarda la matrice dei guadagni dell'osservatore K , essa va progettata opportunamente per riuscire ad ottenere un andamento dell'errore di stima che tenda a zero. Una forma ottimale è conosciuta come “filtro di Kalman”, dal matematico e ingegnere Rudolf E. Kalman famoso per aver ideato e presentato l'algoritmo utile allo scopo. La forma che si ottiene imponendo che la traccia della matrice di covarianza P sia minima (dunque sia minima la somma delle varianze degli elementi dello stato x) è [15]:

$$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1} \quad (4.22)$$

Essendo le matrici P_k ed R definite positive, allora anche la matrice $(H P_k^- H^T + R)$ risulta definita positiva, da cui ne deriva che esiste la sua inversa.

Una volta applicato il filtro, a seguito dell'introduzione della misurazione y_k , si può trovare una forma per ricavare la matrice di covarianza dell'errore di stima dello stato P_k^+ :

$$P_k^+ = [I - K_k H] P_k^- \quad (4.23)$$

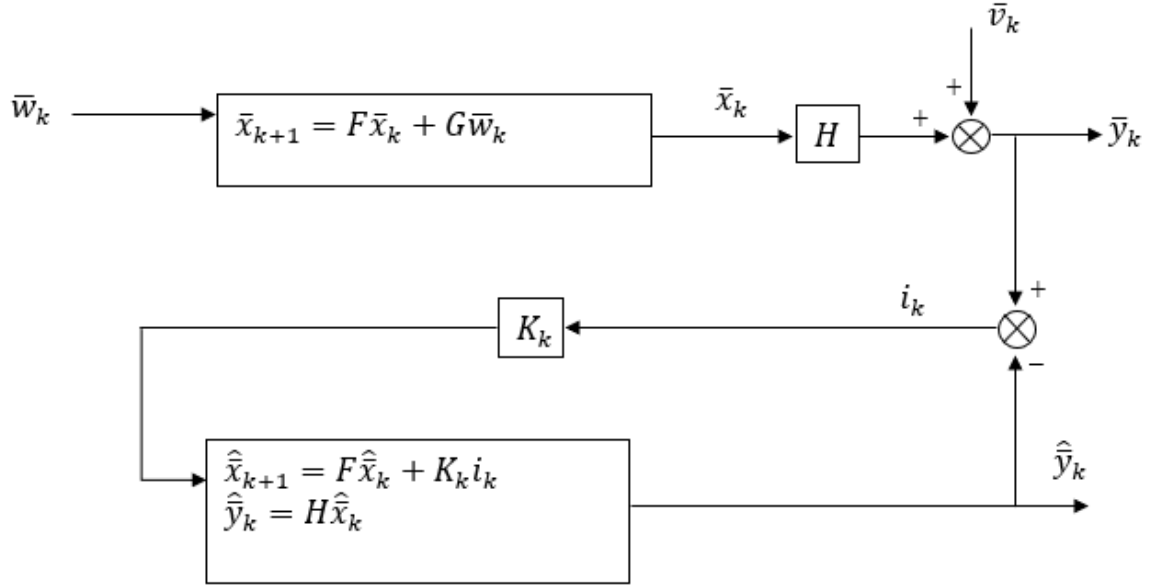


Figura 25: Sistema e Osservatore dello stato

Nei parametri iniziali, seguendo questo modello, andranno dunque specificate le varianze dei disturbi in gioco per impostare le matrici Q ed R , in più dovrà essere inizializzata la matrice di covarianza dell'errore di stima dello stato iniziale P_0 . Si può indicare lo stato $x_k = [p_k^T v_k^T \theta_k^T]^T$ con $p_k \in \mathbb{R}^3$, $v_k \in \mathbb{R}^3$ e $\theta_k \in \mathbb{R}^3$ i vettori degli errori rispettivamente di posizione, velocità e assetto e T_s il periodo di campionamento. I bias dei sensori non vengono per forza inseriti nel vettore dello stato in quanto non sono le maggiori risorse di errore che conducono alla distorsione dell'output dell'IMU durante dinamiche rapide, come accade nel caso di una piattaforma applicata al piede durante la camminata.

Le matrici del modello del sistema sono:

$$F = \begin{bmatrix} I_3 & T_s I_3 & 0_3 \\ 0_3 & I_3 & T_s [a^n]_\times \\ 0_3 & 0_3 & I_3 \end{bmatrix} \quad (4.24)$$

$$G = \begin{bmatrix} 0_3 & 0_3 \\ T_s C_b^n & 0_3 \\ 0_3 & -T_s C_b^n \end{bmatrix} \quad (4.25)$$

$$H = [0_3 \quad I_3 \quad 0_3] \quad (4.26)$$

Nella (4.24), $[a^n]_\times$ è la matrice “skew-symmetric” (antisimmetrica), utilizzata per rappresentare il prodotto vettoriale come moltiplicazione tra matrici, del vettore di accelerazioni espresse nel sistema di navigazione; C_b^n è invece la matrice di rotazione dal sistema in assi body (piattaforma) a quello di navigazione.

Per aggiornare la matrice di covarianza prima dell’applicazione del filtro, vale la seguente relazione:

$$P_{k+1}^- = FP_k^+ F^T + GQG^T \quad (4.27)$$

4.7 Zero-Velocity Detector

Nei paragrafi precedenti è già stato introdotto che per riuscire a migliorare le performance del sistema di navigazione durante la camminata, è necessario sfruttare gli istanti ciclici di stato stazionario del piede per poter correggere gli errori che si sono accumulati nel tempo (figura 15). Per poter implementare questa soluzione è necessario capire come sfruttare i sensori inerziali a disposizione per discriminare i momenti di stato stazionario del piede rispetto agli altri. In questo paragrafo verranno introdotti quattro metodi differenti, per poi scegliere quello che, in base ad analisi già effettuate e descritte in letteratura, consente di ottenere la migliore accuratezza nell’ottenere la posizione durante una camminata in piano a velocità di circa 4-5 km/h [16].

Definendo i seguenti vettori:

$y_n^a \in \mathbb{R}^3$: vettore delle forze specifiche misurate all'istante $n \in \mathbb{N}$

$y_n^\omega \in \mathbb{R}^3$: vettore delle velocità angolari misurate all'istante $n \in \mathbb{N}$

e indicando con $W \in \mathbb{N}$ il numero di istanti di campionamento utilizzati per la rilevazione dello stato stazionario, allora si possono introdurre le seguenti sequenze di misurazioni:

$$z_n^a \triangleq \{y_k^a\}_{k=n}^{n+W-1}$$

$$z_n^\omega \triangleq \{y_k^\omega\}_{k=n}^{n+W-1}$$

Si può dunque esprimere il problema matematico sulla base della soddisfazione o meno di un'ipotesi, con l'ottenimento di un risultato dunque “vero” o “falso”:

$$T(z_n^a, z_n^\omega) < \gamma \quad (4.28)$$

Nella (4.24), $T(z_n^a, z_n^\omega)$ è un risultato statistico che dipende dai valori presenti nelle sequenze di misurazioni e dal tipo di test utilizzato, mentre γ è la soglia impostata.

Indicando con $\sigma_a^2 \in \mathbb{R}$ e $\sigma_\omega^2 \in \mathbb{R}$ rispettivamente le varianze del rumore di misura degli accelerometri e dei giroscopi, introducendo il quadrato del modulo di un vettore come $\|a\|^2 = a^T a$ e definendo la media dei campionamenti:

$$\bar{y}_n^a = \frac{1}{W} \sum_{k=n}^{n+W-1} y_k^a \quad (4.29)$$

si possono elencare e commentare brevemente i quattro metodi di rilevazione del moto stazionario.

The Stance Hypothesis Optimal Detector (SHOE): fa riferimento sia alle misure di accelerazione che a quelle di velocità di rotazione, normalizzate rispetto alla varianza (4.29).

$$T(z_n^a, z_n^\omega) = \frac{1}{W} \sum_{k=n}^{n+W-1} \frac{1}{\sigma_a^2} \left\| y_k^a - g \frac{\bar{y}_n^a}{\|\bar{y}_n^a\|} \right\|^2 + \frac{1}{\sigma_\omega^2} \|y_k^\omega\|^2 \quad (4.30)$$

The Acceleration Moving Variance Detector (MV): si basa sulla variazione dell'accelerazione durante la finestra di riferimento rispetto al valore medio calcolato (4.30).

$$T(z_n^a, z_n^\omega) = \frac{1}{\sigma_a^2 W} \sum_{k=n}^{n+W-1} \|y_k^a - \bar{y}_n^a\|^2 \quad (4.31)$$

The Acceleration Magnitude Detector (MAG): in questo caso l'unico discriminante è il modulo medio dell'accelerazione (ripulita dalla componente gravitazionale) durante la finestra di riferimento (4.31).

$$T(z_n^a, z_n^\omega) = \frac{1}{\sigma_a^2 W} \sum_{k=n}^{n+W-1} (\|y_k^a\| - g)^2 \quad (4.32)$$

The Angular Rate Energy Detector (ARE): l'ultimo metodo si basa solamente sul modulo della velocità di rotazione misurata dai giroscopi della piattaforma (4.32).

$$T(z_n^a, z_n^\omega) = \frac{1}{\sigma_\omega^2 W} \sum_{k=n}^{n+W-1} \|y_k^\omega\|^2 \quad (4.33)$$

È stato dimostrato che i valori singoli delle varianze non incide sulla performance del rilevatore, sebbene per il primo metodo in realtà ci sia un'influenza data dal rapporto $\sigma_a^2/\sigma_\omega^2$, che riflette il confronto dei disturbi nelle misure tra accelerometri e giroscopi. Dagli esperimenti descritti ed illustrati in letteratura, risulta che i metodi maggiormente efficaci siano lo SHOE e l'ARE [16]. Questo è dovuto all'importante ruolo svolto dai giroscopi nella rilevazione del moto stazionario.

Nell'algoritmo è possibile specificare nei parametri iniziali quale metodo si preferisce utilizzare, nonché il valore della soglia γ e il numero di campionamenti che permettono di calcolare il valore di T . Durante l'implementazione, il codice calcola gli elementi del vettore T per tutte le finestre possibili, partendo dai dati IMU inseriti nel vettore u (post-processing), successivamente definisce il vettore "zupt" (dimensionalmente pari a T), ponendo il valore uno ogniqualvolta l'elemento di T è inferiore alla soglia impostata, zero negli altri casi.

4.8 Funzionamento dell'INS con Zero-Velocity Update

In questo paragrafo verranno descritti gli step necessari per il calcolo del percorso effettuato a partire da una matrice contenente i dati degli accelerometri e dei giroscopi per ogni istante di campionamento, insieme ad un vettore che per ogni istante indica se il sistema si trova a velocità zero o meno (in caso affermativo viene applicata la correzione tramite il filtro di Kalman già descritto).

Dapprima vengono inizializzate le matrici H , Q , R e quella iniziale P_0 sulla base dei valori delle varianze inserite nei parametri iniziali. Successivamente viene calcolato lo stato iniziale dell'equazione di navigazione, ovvero gli elementi dei vettori di posizione, velocità e assetto all'istante iniziale. La posizione e la velocità sono considerati nulle, così come l'angolo di heading ψ ; l'angolo di banco φ e quello di elevazione ϑ iniziali vengono calcolati sotto

l'ipotesi che il sistema rimanga stazionario durante i primi istanti di campionamento: si procede ricavando la media dei dati degli accelerometri di questi primi campionamenti (ad esempio venti), per poi sfruttare i risultati con le seguenti relazioni, ricavabili a partire dalla (2.11):

$$\begin{cases} \vartheta_0 = \arctan2(\bar{a}_x, \sqrt{\bar{a}_y^2 + \bar{a}_z^2}) \\ \varphi_0 = \arctan2(\bar{a}_y, \bar{a}_z) \end{cases} \quad (4.34)$$

dove $\bar{a} = [\bar{a}_x, \bar{a}_y, \bar{a}_z]^T$ è il vettore delle accelerazioni medie calcolate.

Seguendo la (2.8) viene calcolata la matrice di rotazione C_{NED}^{body} , utilizzata poi per calcolare i quaternioni che corrispondono all'assetto iniziale.

Nota: I quaternioni

I quaternioni furono introdotti da R. Hamilton intorno alla metà del XIX secolo per generalizzare i numeri complessi allo spazio tridimensionale, e vengono indicati generalmente con la lettera q. Essi permettono di rappresentare in maniera conveniente orientamenti e rotazioni di oggetti in 3 dimensioni; inoltre sono più stabili numericamente rispetto alle matrici di rotazioni, il che li rende preferibili a livello computazionale.

Il quaternione è un elemento dello spazio lineare $\mathbb{H}(\mathbb{R})$ di quattro dimensioni, definito sul campo dei numeri reali \mathbb{R} con base $\{1 \ i \ j \ k\}$, e valgono le seguenti relazioni [17]:

$$\begin{aligned} i^2 &= j^2 = k^2 = ijk = -1 \\ ij &= -ji = k \\ jk &= -kj = i \\ ki &= -ik = j \end{aligned} \quad (4.35)$$

Il quaternione è dunque espresso tramite una combinazione lineare nella base indicata:

$$q = q_0 + q_1i + q_2j + q_3k \quad (4.36)$$

in cui i coefficienti q_i con $i = 0,1,2,3$ sono numeri reali. Analogamente ai numeri complessi, il quaternion è rappresentabile con un vettore quadruplo $q = (q_0 \ q_1 \ q_2 \ q_3)^T$.

In particolare, possiamo introdurre i quaternioni unitari, tali per cui $\|q\| = 1$, e possono anche essere scritto in questo modo:

$$q = \cos\theta + \mathbf{u}\sin\theta = e^{\mathbf{u}\theta} \quad (4.37)$$

in cui \mathbf{u} è un vettore di norma unitaria e θ un angolo generico. Questo quaternion rappresenta la rotazione di un angolo 2θ intorno all'asse rappresentato dal versore $\mathbf{u} = (u_1 \ u_2 \ u_3)^T$; perciò per esprimere la rotazione di un angolo θ attorno all'asse di versore \mathbf{u} il quaternion sarà:

$$q = \cos\frac{\theta}{2} + \mathbf{u}\sin\frac{\theta}{2} \quad (4.38)$$

Poiché sappiamo già che una rotazione può essere espressa tramite una matrice quadrata 3x3 (che è ortonormale e possiede determinante positivo pari a +1), è opportuno capire quali siano le relazioni necessarie per ottenere il quaternion dalla matrice di rotazione stessa e viceversa.

Per passare dalla matrice R con elementi r_{ij} al corrispondente quaternion q , valgono le seguenti relazioni:

$$\begin{aligned} q_0 &= \mp \frac{1}{2} \sqrt{(1 + r_{11} + r_{22} + r_{33})} \\ q_1 &= \frac{1}{4q_0} (r_{32} - r_{23}) \\ q_2 &= \frac{1}{4q_0} (r_{13} - r_{31}) \\ q_3 &= \frac{1}{4q_0} (r_{21} - r_{12}) \end{aligned} \quad (4.39)$$

L'algoritmo implementato fa riferimento alla (4.38), però presenta anche uno

studio sul valore della traccia della matrice di rotazione e, nel caso questo sia prossimo allo zero o minore di zero, le relazioni vengono leggermente modificate in base agli elementi presenti sulla diagonale principale (si individua quello che ha valore maggiore), anche per evitare grandi errori numerici. Per una descrizione più dettagliata riferirsi a [18].

Inversamente, per passare da un quaternionione q alla matrice di rotazione corrispondente R , si applica la relazione:

$$R(q) = \begin{pmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_3q_0) & 2(q_1q_3 + q_2q_0) \\ 2(q_1q_2 + q_3q_0) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 - q_1q_0) \\ 2(q_1q_3 - q_2q_0) & 2(q_2q_3 + q_1q_0) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{pmatrix} \quad (4.40)$$

Fine Nota

Dopo aver ricavato tutte le condizioni iniziali necessarie, viene avviato l'algoritmo principale per effettuare l'integrazione dei dati IMU. Nel caso in cui lo stato comprenda l'evoluzione dell'errore dovuto al bias dei sensori ed anche quello dovuto al fattore di scala (descritti entrambi al paragrafo 2.5), viene applicata una compensazione in relazione ad entrambi, in modo da ottenere dei dati "ripuliti" da utilizzare nei calcoli successivi.

Denominando $\mathbf{f} = [f_1 \dots f_6]^T$ il vettore contenente le sei componenti del fattore di scala (ogni componente è relativo ad un sensore della piattaforma) ed allo stesso modo $\mathbf{b} = [b_1 \dots b_6]^T$ il vettore con le sei componenti del bias, vale la seguente relazione:

$$\mathbf{u}_{out} = \begin{pmatrix} 1 & \dots & 0 \\ \frac{1}{1-f_1} & \dots & \vdots \\ \vdots & \ddots & 1 \\ 0 & \dots & \frac{1}{1-f_6} \end{pmatrix} \mathbf{u}_{in} + \mathbf{b} \quad (4.41)$$

dove u_{in} e u_{out} sono i vettori contenenti i dati da accelerometri e giroscopi rispettivamente prima e dopo la compensazione di fattori di scala e bias.

Successivamente vengono applicate le equazioni di navigazione relative ad un sistema inerziale: in questo caso sono stati trascurati i termini oltre un certo grado, dunque sono valide solo dove ci si aspetta una velocità moderata.

Indicando con $\bar{\omega} = (\omega_1 \ \omega_2 \ \omega_3)^T$ il vettore delle velocità angolari relative ad un certo istante di campionamento, allora si definiscono le seguenti variazioni angolari:

$$\begin{cases} p = \omega_1 T_s \\ q = \omega_2 T_s \\ r = \omega_3 T_s \end{cases} \quad (4.42)$$

Dalla (4.41), in cui T_s è il periodo di campionamento, possiamo scrivere la matrice di aggiornamento dei quaternioni:

$$\Omega = \frac{1}{2} \begin{bmatrix} 0 & r & -q & p \\ -r & 0 & p & q \\ q & -p & 0 & r \\ -p & -q & -r & 0 \end{bmatrix} \quad (4.43)$$

Si definisce $v = \|\bar{\omega}\|T_s$, che indica l'entità della rotazione, e nel caso in cui il suo valore sia diverso da zero, allora il quaternioni viene aggiornato tramite la seguente relazione:

$$q_{out} = \left(\cos\left(\frac{v}{2}\right) I_4 + \frac{2}{v} \sin\left(\frac{v}{2}\right) \Omega \right) q_{in} \quad (4.44)$$

Infine, si divide il quaternione ottenuto per il suo modulo, per essere sicuri che il suo modulo sia unitario (dunque che rappresenti una rotazione tridimensionale) e compensare eventuali errori numerici.

Col quaternione aggiornato possiamo calcolare la matrice di rotazione dal sistema assi body a quello di navigazione attraverso la (4.39), e si ottiene una matrice che può essere anche espressa in questo modo:

$$C_b^n = \begin{pmatrix} c \vartheta c \psi & s \vartheta s \varphi c \psi - c \varphi s \psi & s \vartheta c \varphi c \psi + s \varphi s \psi \\ c \vartheta c \vartheta s \psi & s \vartheta s \varphi s \psi + c \varphi c \psi & s \vartheta c \varphi s \psi - s \varphi c \psi \\ -s \vartheta & s \varphi & c \varphi c \vartheta \end{pmatrix} \quad (4.45)$$

Dalla (4.45), omettendo momentaneamente apice e pedice della matrice per alleggerire la notazione, risulta immediato ottenere l'assetto espresso negli angoli di Eulero:

$$\begin{cases} \varphi = \text{atan2}(C_{32}, C_{33}) \\ \vartheta = -\text{atan}\left(C_{31}, \sqrt{1 - C_{31}^2}\right) \\ \psi = \text{atan2}(C_{21}, C_{11}) \end{cases} \quad (4.46)$$

A questo punto occorre calcolare le accelerazioni lineari (dunque sottraendo la componente gravitazionale) nel sistema di navigazione. Definendo \bar{a}_l il vettore che vogliamo calcolare, \bar{a} il vettore di accelerazioni date dalla piattaforma IMU e $\bar{g} = (0 \ 0 \ -g)^T$ il vettore di accelerazione gravitazionale nel sistema di navigazione, allora vale:

$$\bar{a}_l = C_b^n \bar{a} + \bar{g} \quad (4.47)$$

Definendo le seguenti matrici:

$$A = \begin{bmatrix} 1 & 0 & 0 & T_s & 0 & 0 \\ 0 & 1 & 0 & 0 & T_s & 0 \\ 0 & 0 & 1 & 0 & 0 & T_s \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.5T_s^2 & 0 & 0 \\ 0 & 0.5T_s^2 & 0 \\ 0 & 0 & 0.5T_s^2 \\ T_s & 0 & 0 \\ 0 & T_s & 0 \\ 0 & 0 & T_s \end{bmatrix}$$

si può scrivere la seguente relazione per il calcolo dello della posizione e velocità aggiornate rispetto a quelle precedenti e rispetto alle accelerazioni lineari già ricavate:

$$\bar{x}_{output} = A\bar{x}_{input} + B\bar{a}_l \quad (4.48)$$

Nella (4.47), \bar{x}_{output} e \bar{x}_{input} sono vettori con sei elementi ciascuno, ovvero le componenti di posizione e velocità, rispettivamente dopo e prima l'aggiornamento effettuato attraverso i dati a disposizione.

Successivamente vengono calcolate la matrice di transizione F e quella del rumore di processo G come descritto nella (4.24) e (4.25). Attraverso queste, si ricava la matrice di covarianza P^+ secondo la (4.26).

Il prossimo passo prevede la lettura dell'informazione nel vettore "zupt" (contiene il valore uno se viene rilevata velocità nulla, e zero viceversa), e nel caso lo stato in un certo istante di campionamento sia stazionario allora si applica la correzione col filtro di Kalman. Innanzitutto, viene calcolata la matrice dei guadagni dell'osservatore secondo la (4.22), per poi ricavare il valore dell'innovazione i , che equivale comunque all'opposto del valore della velocità stimata tramite l'integrazione, poiché secondo il rilevatore dovrebbe essere nulla:

$$i = v - \hat{v} = 0 - \hat{v} \quad (4.49)$$

Moltiplicando l'innovazione per la matrice dei guadagni calcolata K si calcola l'entità della correzione che dev'essere applicata allo stato. La posizione e la velocità correnti sono corrette direttamente con la perturbazione trovata seguendo la (4.17). Per la correzione dell'assetto si convertono i quaternioni non corretti nella matrice di trasformazione corrispondente e, chiamando $\bar{\varepsilon}$ il vettore con le tre correzioni dell'assetto (ultime tre componenti del vettore di stato della stima delle perturbazioni), e costruendo la seguente matrice

$$Z = \begin{bmatrix} 0 & -\varepsilon_3 & \varepsilon_2 \\ \varepsilon_3 & 0 & -\varepsilon_1 \\ -\varepsilon_2 & \varepsilon_1 & 0 \end{bmatrix}$$

si può dunque correggere la matrice di rotazione R attraverso la seguente relazione:

$$R^+ = (I_3 - Z)R^- \quad (4.50)$$

Dalla matrice di rotazione aggiornata con la correzione dello stato, è possibile risalire sia ai quaternioni corretti che all'assetto espresso in angoli di Eulero. Infine, viene anche aggiornata la matrice di covarianza P con la (4.23).

Dopo aver iterato il procedimento per ogni istante di campionamento, si arriva alla posizione finale calcolata che rappresenta direttamente l'errore, avendo impostato le coordinate iniziali come nulle ed essendo i dati a disposizione relativi ad un circuito chiuso.

4.9 Grafici e Valutazione dell'Algoritmo

Inserendo un particolare set di dati messi a disposizione online dal progetto OpenShoe, dall'algoritmo descritto sono stati ottenuti i seguenti grafici attraverso i calcoli che seguono il modello teorico esposto:

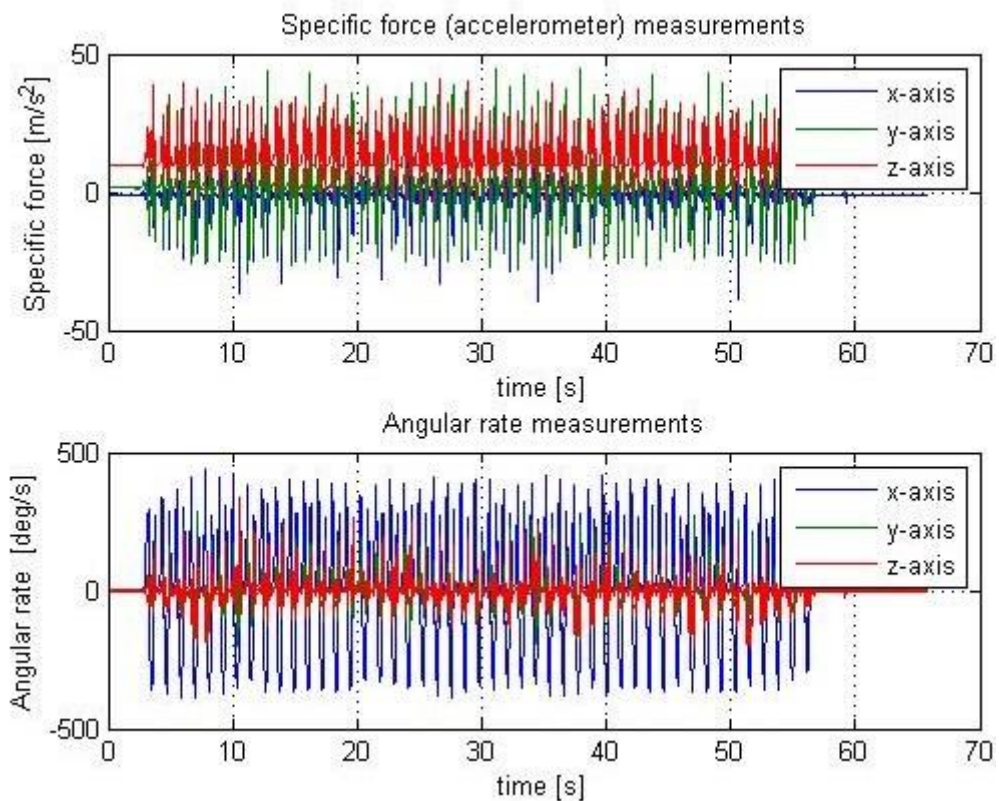


Figura 26: Dati dalla piattaforma IMU

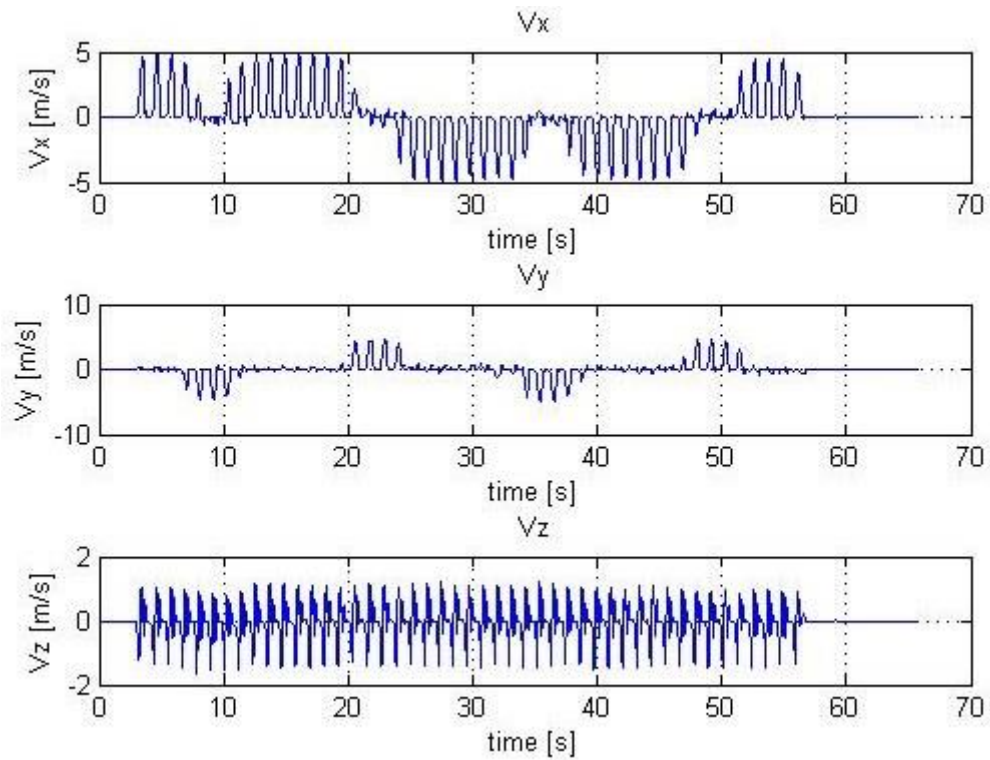


Figura 27: Velocità calcolate lungo i tre assi

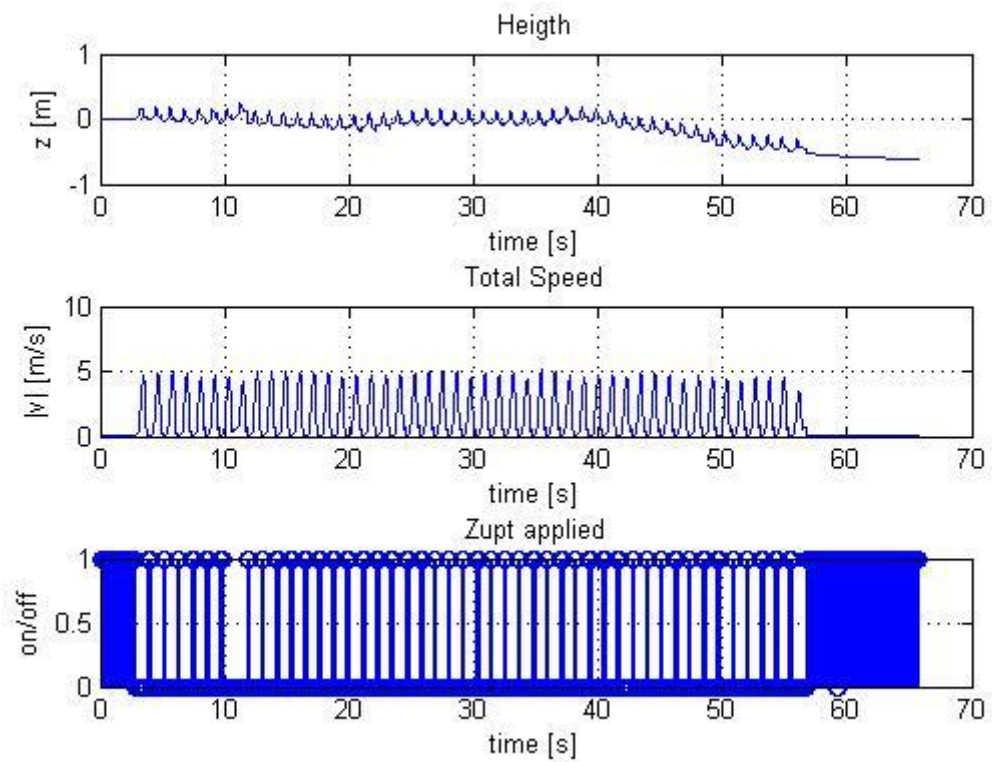


Figura 28: Andamento della coordinata verticale, della velocità totale e applicazione o meno del ZUPT

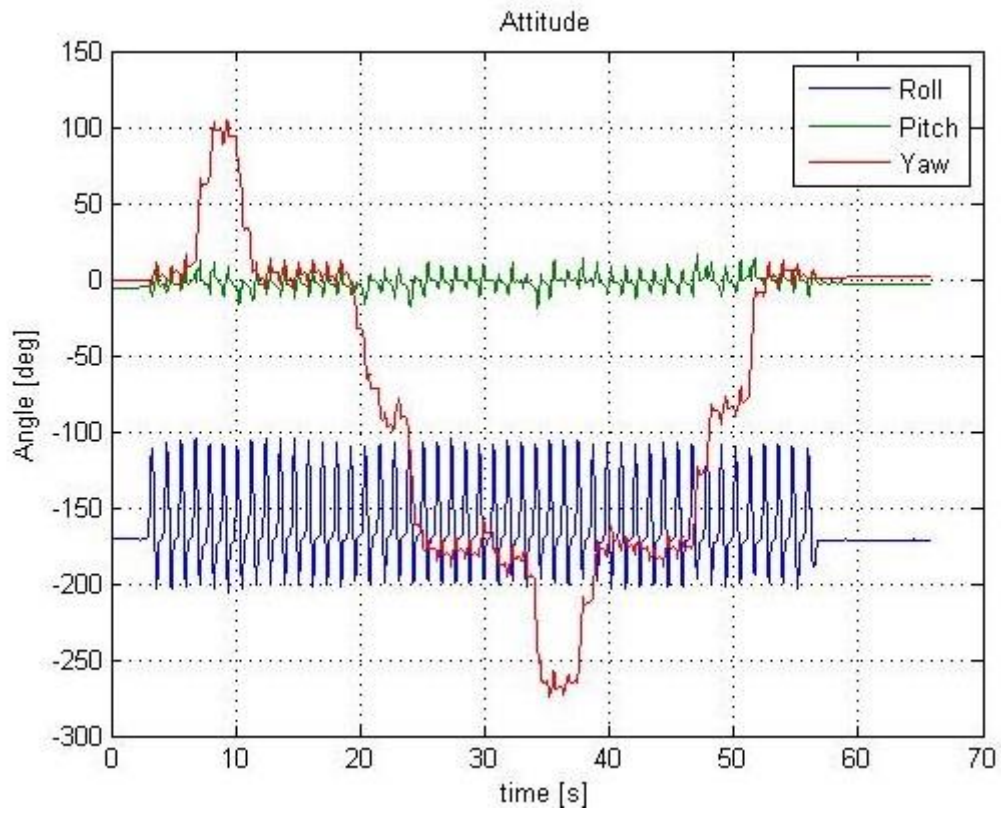


Figura 29: Andamento dell'assetto (angoli di Eulero)

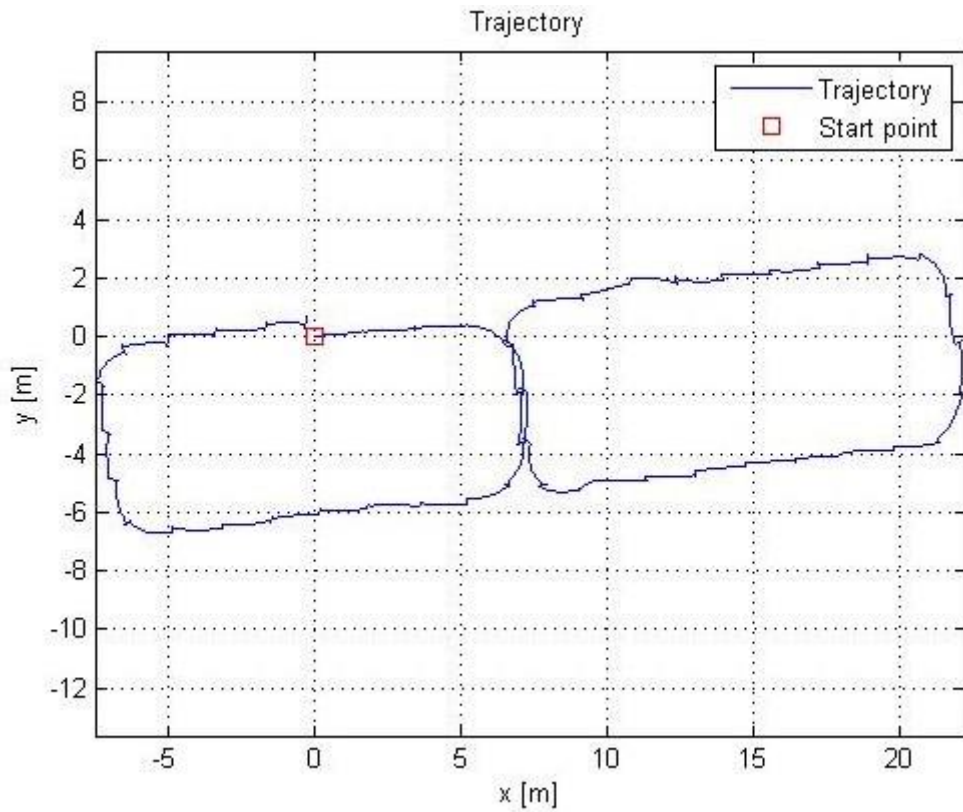


Figura 30: Traiettorie nel piano

Attraverso i calcoli, si ottiene un errore orizzontale (nel piano) di circa 0.7m ed un errore sferico (comprendendo la coordinata verticale) di circa 0.93m, che corrispondono in un percorso stimato intorno agli 80m [19], ad errori in percentuale rispettivamente di 0.875% e di 1.16%. La performance dipende anche dal tipo di traiettoria seguita: l'ostacolo principale è dato dal fatto che gli errori di posizione sono fortemente accoppiati con quelli di assetto attraverso la reale posizione relativa. Per questo motivo, la traiettoria a circuito chiuso presenta dei vantaggi nei risultati ottenuti, e in esperimenti futuri sarebbe opportuno provare ad ottenere i dati relativi ad una traiettoria rettilinea per poi effettuare un confronto. Un altro aspetto che influenza l'errore è sicuramente l'intervallo di tempo del moto, infatti se la stessa distanza venisse percorsa correndo, l'errore diminuirebbe per il fatto che si accumulano meno disturbi nelle misurazioni dei giroscopi.

Si nota che dalla figura 27 è possibile capire quali sono gli istanti in cui viene cambiata la direzione della camminata, in quanto le componenti x e y della velocità subiscono forti variazioni; dalla figura 28 è poi possibile stimare il numero di passi effettuati, poiché è necessario solamente contare il numero di volte in cui viene applicato lo ZUPT (escludendo gli intervalli iniziale e finale), che rappresenta il numero di cicli del singolo piede, e raddoppiarne il valore.

5 Analisi dei Dati

5.1 Obiettivo dell'Analisi

Il progetto OpenShoe già citato e descritto, oltre a mettere a disposizione gli algoritmi progettati per gli esperimenti a scopo di libero studio, permette di accedere ai dati della piattaforma IMU utilizzata durante le verifiche, in particolare una MicroStrain 3DX-GX2, le cui caratteristiche tecniche principali sono già state presentate. I vari set di dati a disposizione sono legati al medesimo circuito chiuso “a otto” percorso, con piccole variazioni tra uno e l'altro a causa di circostanze contingenti. È stato pensato di effettuare un'analisi dei segnali attraverso la trasformata di Fourier ma ancora più in particolare ricavando il grafico della densità spettrale di potenza, in modo da poter verificare alcune proprietà della camminata ed anche dedurre alcune modalità per poter rilevare più rapidamente il cambio di direzione durante il moto.

5.2 Trasformata di Fourier

Nel campo dello studio dei segnali, talvolta si ritiene opportuno passare dall'analisi nel dominio del tempo a quella nel dominio delle frequenze, dato che quest'ultima permette in certi casi di trovare una risoluzione ad un problema o semplicemente ricavare informazioni ulteriori sul segnale. Lo strumento matematico che permette di effettuare questa particolare analisi è la trasformata di Fourier, sviluppata dal matematico francese Jean Baptiste Joseph Fourier nella prima metà del XIX secolo.

Data ad esempio una funzione $x(t)$ continua nel tempo, la sua trasformata di Fourier $X(f)$ è per definizione [20]:

$$X(f) \triangleq \int_{-\infty}^{+\infty} x(t)e^{-j2\pi ft} dt \quad (5.1)$$

Con la (5.1) viene dunque rappresentato un segnale continuo come somma di infiniti esponenziali periodici (j è l'unità immaginaria). Risulta possibile anche il processo inverso, ovvero tornare al dominio del tempo tramite l'antitrasformata:

$$x(t) = \int_{-\infty}^{+\infty} X(f)e^{+j2\pi ft} df \quad (5.2)$$

La rappresentazione della trasformata avviene tramite gli spettri di ampiezza e di fase. La (5.1) non è però implementabile in un sistema digitale, innanzitutto perché richiede segnali continui nel tempo, inoltre per il fatto che l'integrale comporterebbe un numero infinito di dati da elaborare; sono altresì necessarie tre operazioni fondamentali: il campionamento, il troncamento del segnale e la discretizzazione dell'asse delle frequenze.

La conseguenza principale del campionamento è che, dato un segnale continuo ed il suo spettro di Fourier, allora lo spettro del corrispondente segnale campionato con frequenza f_{camp} è costituito dalle repliche dello spettro originale, traslate su frequenze multiple della frequenza di campionamento f_{camp} (figura 31).

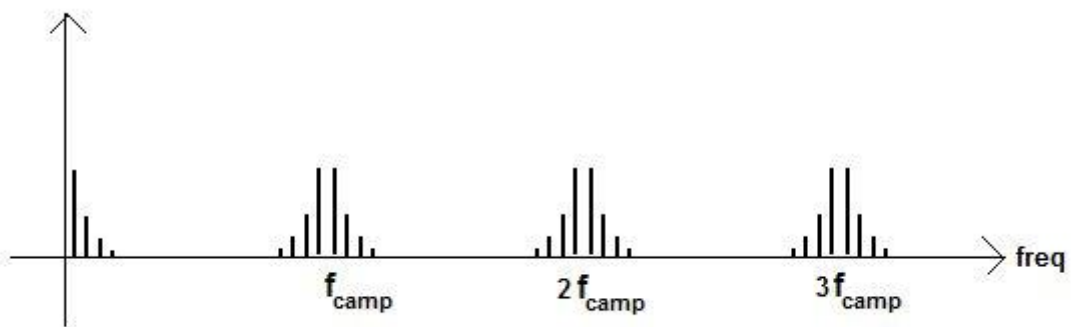


Figura 31: Spettro di un segnale campionato

Per evitare dunque sovrapposizioni tra le repliche, è necessario che la frequenza di campionamento sia maggiore o uguale all'intervallo della banda di

frequenze che costituiscono lo spettro del segnale continuo, altrimenti vi sarà una sovrapposizione (teorema di campionamento o di Shannon). In altre parole, per una corretta analisi in frequenza è opportuno che il segnale sia a spettro limitato (tale banda prende il nome di “frequenza di Nyquist”) e che la frequenza di campionamento sia la più elevata possibile. Se ciò non accade, la sovrapposizione renderà impossibile distinguere tra lo spettro originario e le repliche (aliasing), causando così una non corretta ricostruzione del segnale.

Il troncamento è dovuto al fatto che il segnale avrà in generale un inizio ed una fine nel tempo, e può essere pensato come un segnale infinito prelevato attraverso un’opportuna finestra temporale (window):

$$x_w(t) = x(t)w(t) \quad (5.3)$$

Tale osservazione in un intervallo temporale limitato comporta che, se lo spettro del segnale originario conteneva delle transizioni nette, come nel caso di segnali periodici, tali transizioni vengono smussate tanto più quanto più è stretta la finestra di campionamento.

Infine, la trasformata di Fourier non può essere valutata per tutti i valori della variabile f , e inoltre basterebbe conoscere l’andamento dello spettro solamente nell’intervallo $[0, f_{camp}]$, motivo per cui è stata introdotta la trasformata discreta di Fourier (DFT), che permette di analizzare il contenuto armonico in quell’intervallo tramite le N componenti discrete. Chiamando f_w la risoluzione in frequenza della DFT, ovvero il valore base di cui tutte le frequenze in cui viene valutato lo spettro sono multiple, allora il legame di trasformazione tra gli N campioni nel tempo $x_i = x(iT_s)$ e i campioni in frequenza $X_k = X(kf_w)$ è:

$$X_k = \sum_{i=0}^{N-1} x(iT_s) e^{-j2\pi k f_w i T_s} = \sum_{i=0}^{N-1} x_i e^{-j\frac{2\pi}{N} k i} \quad (5.4)$$

dove $f_w = f_{camp}/N$.

5.3 La funzione Periodogram in MATLAB

Per l'analisi in frequenza dei dati a disposizione è stata scelta la funzione Periodogram disponibile in MATLAB, che permette di calcolare la densità spettrale di potenza (PSD) di un segnale discreto. Per definizione, dato un generico segnale di potenza $x(t)$ con trasformata di Fourier $X(f)$, allora la densità spettrale di potenza è data dalla seguente funzione [21]:

$$P_{xx}(f) \triangleq \lim_{T \rightarrow +\infty} \left(\frac{|X_T(f)|^2}{T} \right) \quad (5.5)$$

in cui $X_T(f) := F\{x_T(t)\}$ è la trasformata di Fourier del segnale troncato $x_T(t)$ all'interno della finestra temporale $(-T/2, T/2)$:

$$x_T(t) := \begin{cases} x(t), & \text{se } 0 \leq |t| \leq T/2 \\ 0, & \text{se } |t| > T/2 \end{cases} \quad (5.6)$$

La funzione Periodogram sfrutta la trasformata di Fourier veloce (Fast Fourier Transform, FFT) del segnale, un algoritmo ottimizzato per evitare le N^2 operazioni necessarie nella DFT: ad esempio la *radix-2 FFT* consente di ridurre il numero di operazioni a $N \log_2 N$.

La stima del PSD di un segnale $x_L(n)$ di lunghezza L viene ricavata con la seguente relazione [22]:

$$P_{xx}(f) = \frac{1}{LF_s} \left| \sum_{n=0}^{L-1} x_L(n) e^{-j2\pi f n / F_s} \right|^2 \quad (5.7)$$

Nella (5.7) F_s è la frequenza di campionamento.

5.4 Analisi dei Dati IMU

La forma utilizzata per l'analisi dei dati IMU con la funzione Periodogram descritta è la seguente:

$$[p_{xx}, f] = \text{periodogram}(yy, [\quad], f, f_s) \quad (5.8)$$

Nella (5.8) yy è il vettore contenente i valori del segnale da analizzare in frequenza, f è il vettore delle frequenze per le quali si ha interesse ottenere il calcolo, f_s è la frequenza di campionamento, mentre il secondo valore lasciato di default è il tipo di finestra utilizzata dalla funzione (in questo caso di default è rettangolare, dunque che non attenua il peso dei valori del segnale agli estremi).

Il vettore delle frequenze f è stato impostato da 0 Hz a 5 Hz. In realtà, in particolare per gli accelerometri, è stato inizializzato il vettore delle frequenze da 0.05 Hz, per ottenere grafici in scala più comodi da analizzare eliminando la componente costante attorno a cui oscilla il segnale (frequenza zero).

Considerando che la piattaforma IMU è montata nella suola in modo che l'asse 1 sia quello laterale rispetto al piede (beccheggio), l'asse 2 quello longitudinale (rollio) e l'asse 3 quello verticale (imbardata), si ottengono le seguenti densità spettrali di potenza relative ai singoli sensori e al modulo complessivo:

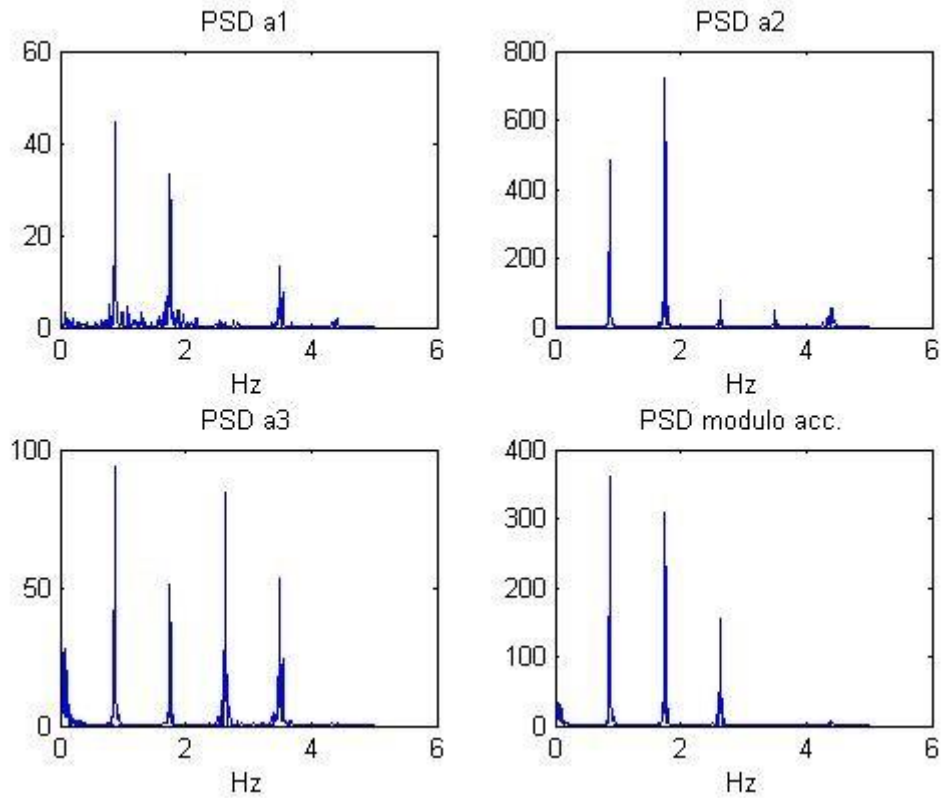


Figura 32: PSD degli accelerometri singoli e del modulo dell'accelerazione

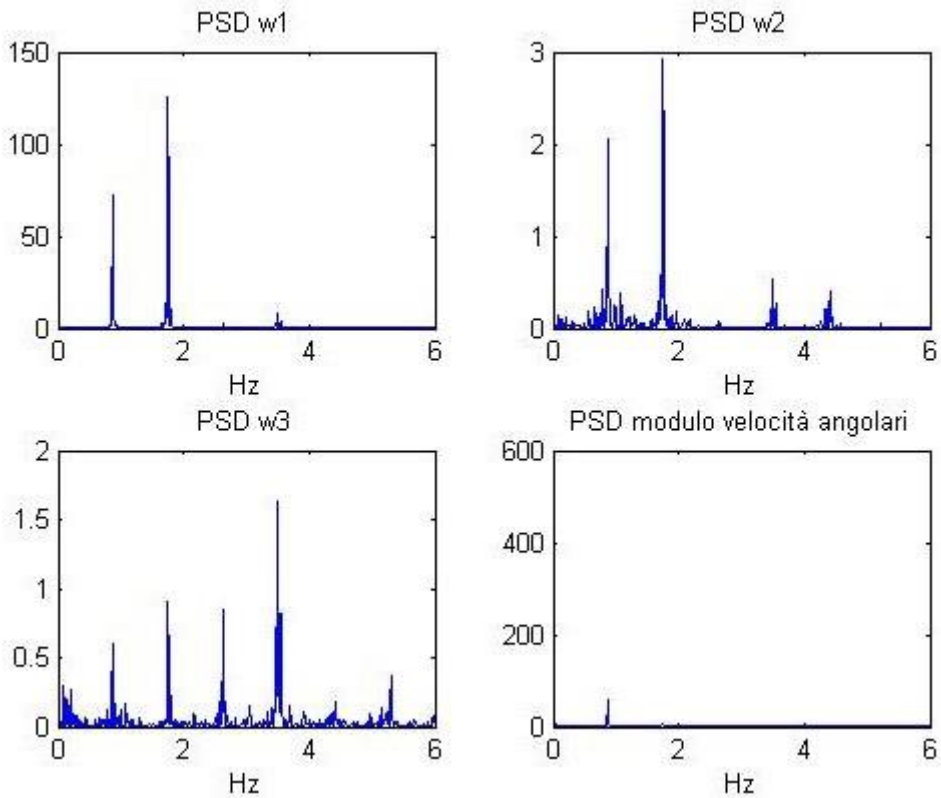


Figura 33: PSD dei singoli giroscopio e del modulo della velocità angolare

Ciò che innanzitutto salta all'occhio è la presenza di picchi a frequenze analoghe per tutti i sensori: in particolare il primo picco evidente viene ottenuto a circa $0.85 \div 0.86$ Hz, dopodiché quelli successivi corrispondono a frequenze multiple della prima. Per analizzare e spiegare meglio questo risultato, è stato studiato il grafico della figura 28, contando il numero di volte n in cui è stato applicato lo ZUPT, escludendo gli estremi in quanto indicatori di una fase precedente e seguente alla camminata in sé. Questo numero (46 per la particolare misurazione mostrata) è indicatore del numero di cicli effettuati dallo stesso piede, dove ogni ciclo corrisponde a due passi. Considerando un Δt di camminata approssimativo (sempre dallo stesso grafico) pari a circa 54 s, allora si può così calcolare la durata media del ciclo:

$$\bar{T}_c = \frac{\Delta t}{n} \approx 1.174 \text{ s} \quad (5.9)$$

da cui si ottiene una frequenza media del ciclo pari a:

$$\bar{f}_c = \frac{1}{\bar{T}_c} \approx 0.852 \quad (5.10)$$

Confrontando il risultato della (5.10) con i grafici illustrati in questo paragrafo, si nota che l'analisi in frequenza è un'ulteriore verifica della periodicità dinamica cui è sottoposto il piede durante la camminata.

La presenza di picchi anche a frequenza multiple rispetto a quella del ciclo è spiegabile con il fenomeno dell'aliasing. Nel momento in cui si procede con un'analisi in frequenza di un segnale campionato, dunque di una funzione non continua nel tempo ma con valori che si susseguono ad intervalli discreti, esso può essere rappresentato da una funzione periodica di una certa frequenza, ma anche da funzioni periodiche di frequenze multiple alla prima, come illustrato in figura 34. In essa si nota che se la frequenza di campionamento è pari a 4 kHz (punti neri), allora una sinusoidale che può approssimare il segnale discreto possiede una frequenza $f_1 = 0.5 \text{ kHz}$, ma un'altra sinusoidale che può approssimarla possiede una frequenza multipla $f_2 = 3.5 \text{ kHz}$. Se invece si raddoppiasse la frequenza di campionamento, non andrebbero più bene entrambe le sinusoidi.

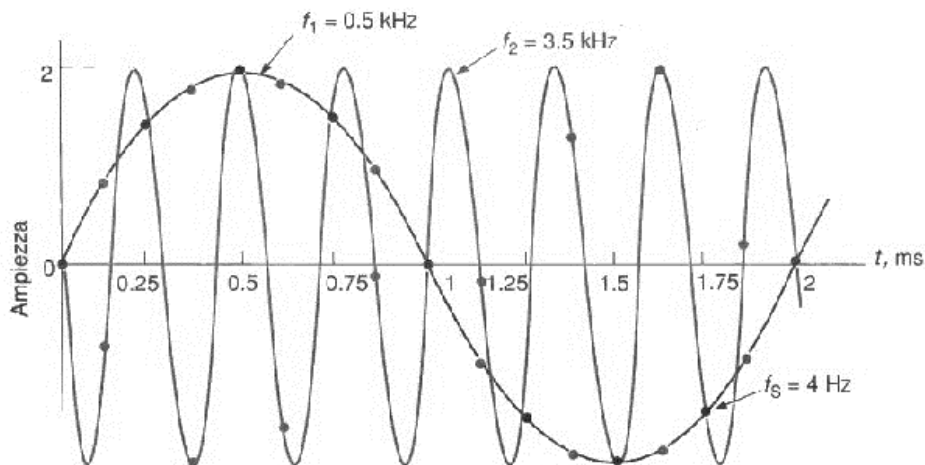


Figura 34: Fenomeno dell'aliasing

Un altro aspetto che si può notare dall'analisi in frequenza dei segnali, è la diversa ampiezza di oscillazione a cui sono sottoposti i vari sensori. Osservando in particolare l'ordinata dei grafici nelle figure 32 e 33 in corrispondenza della frequenza in cui è presente il primo picco, si può dedurre che l'accelerometro che è sottoposto a sollecitazioni maggiori è quello longitudinale, il che è comprensibile data la forte accelerazione e decelerazione cui è sottoposto il piede in tale direzione durante la camminata; tra i giroscopi le oscillazioni maggiori sono presenti in quello di beccheggio a tale frequenza.

5.5 Rilevamento del Cambio di Orientamento

Un aspetto interessante che vale la pena di approfondire è la possibilità di accorgersi del cambio di direzione del piede durante la camminata attraverso l'analisi in frequenza di alcuni sensori, non essendo possibile accorgersi solamente dai grafici di figura 26, che mostrano l'andamento dei vari sensori nel tempo, di questo fenomeno.

Per capire il range di frequenze in cui occorre studiare la densità spettrale di potenza dei segnali, ipotizzando che i cambi di direzione (figura 30) effettuati siano distribuiti uniformemente nel tempo, si calcola il rapporto tra il tempo totale di camminata ed il numero di curve, per ottenere l'intervallo di tempo

che intercorre tra una curva e l'altra. Ricavando il reciproco di questo valore, si giunge ad una frequenza approssimativa che dà l'idea dell'ordine di grandezza attorno a cui concentrarsi per rilevare il cambio di orientamento. In questo caso conviene concentrarsi nell'intervallo compreso tra 0.1 Hz e 0.2 Hz.

Innanzitutto è stato osservato più nel dettaglio la PSD di tutti gli accelerometri, poi posti a confronto, e analogamente di tutti i giroscopi nel range di frequenze individuato, ed è stato notato che mentre per i primi non c'è uno dei sensori che spicchi rispetto agli altri (motivo per cui non ne vengono mostrati i grafici), per i giroscopi è evidente come quello di imbardata principalmente, e quello di rollio in parte, abbiano un'ampiezza che potrebbe denotare un cambio di direzione (figura 35).

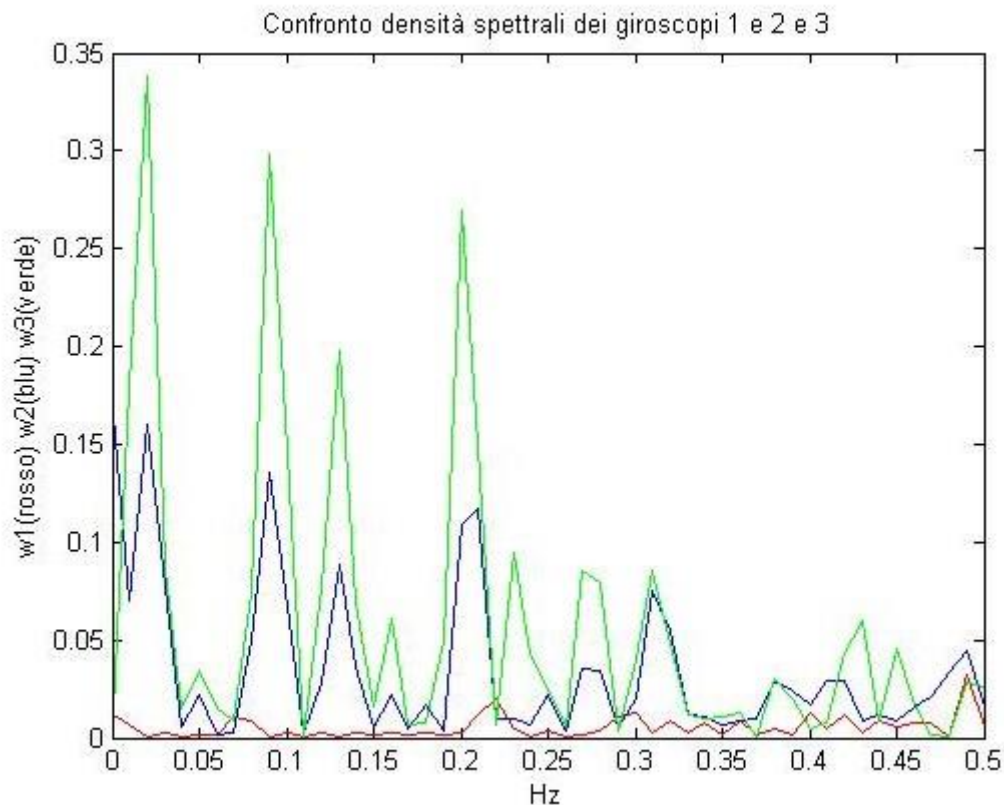


Figura 35: Confronto tra le PSD dei tre giroscopi

Ciò comunque non basta da solo a verificare che quelle ampiezze a tali frequenze siano indice del cambio di orientamento. Si è deciso dunque di utilizzare i dati IMU a disposizione isolando un intervallo temporale in cui, in base

ai calcoli e alla figura 27, non ci siano state curve, per poi effettuare nuovamente un'analisi in frequenza e confrontare i risultati ottenuti. L'intervallo selezionato è quello che va da 11 a 19 secondi, e calcolando la densità spettrale di potenza per i giroscopi in questo tratto per poi parlo a confronto con quella del percorso completo, si ottengono i grafici in figura 36.

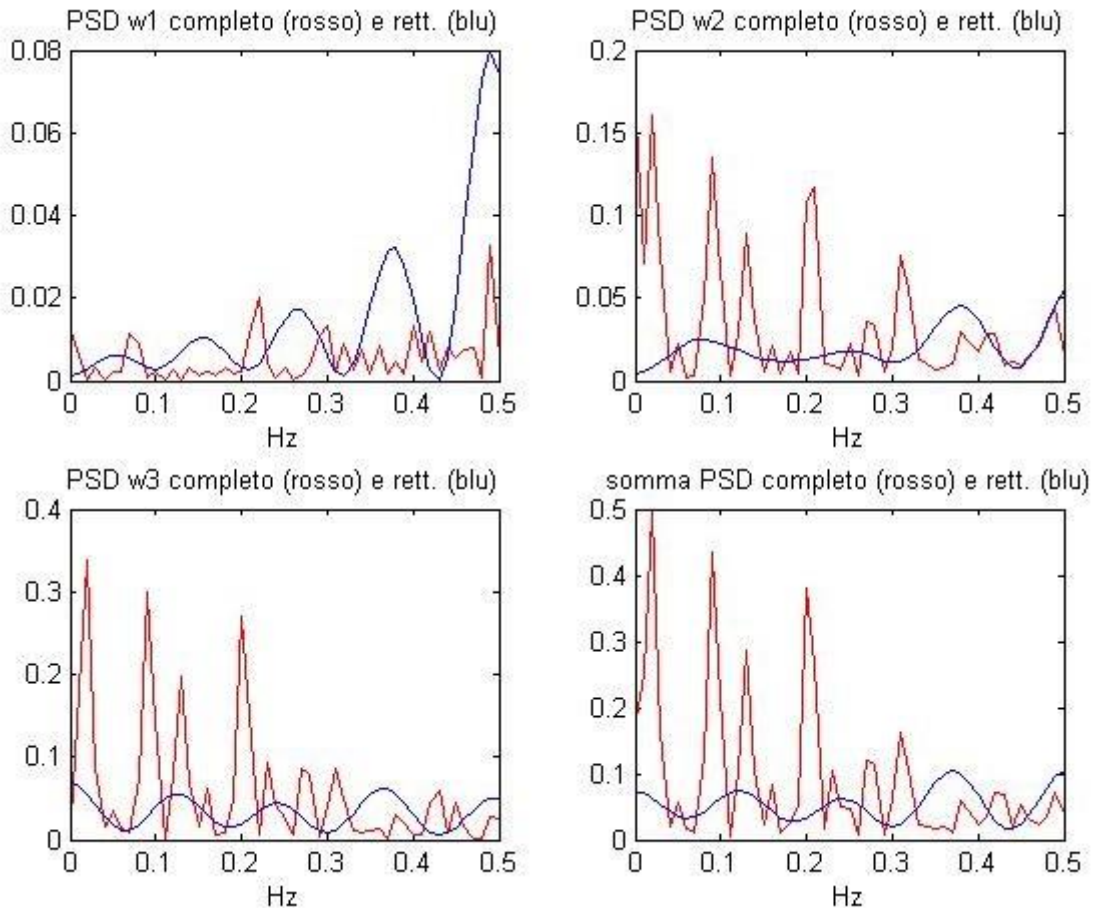


Figura 36: Confronto tra le PSD del percorso completo (rosso) e del tratto rettilineo (blu) per i giroscopi

Si può notare come mentre per il giroscopio 1 non c'è una differenza rilevante, per gli altri due invece è ben evidente il distacco di ampiezza alle basse frequenze tra il percorso completo ed il tratto rettilineo; il risultato è ulteriormente corroborato dal quarto grafico, che è stato ricavato dalla somma dei primi tre.

Poiché il giroscopio di imbardata è quello che (anche intuitivamente) rileva maggiormente il cambio di direzione alle basse frequenze, è stato applicato al

segnale ad esso relativo un filtro “passa-basso” per isolare le oscillazioni di frequenze al di sotto della soglia $f = 0.3 \text{ Hz}$, eliminando tutte quelle di frequenza superiore. Come primo studio è stato realizzato un filtro del primo ordine analogamente al capitolo 3. In particolare, la pulsazione di rottura del filtro sarà dunque $\omega = 2\pi f$, da cui è semplice risalire alla costante di tempo:

$$\tau = \frac{1}{\omega} \quad (5.11)$$

Chiamando x il segnale di partenza, x_f il segnale filtrato e dt il periodo di campionamento, vale la relazione:

$$x_f(t + dt) = x_f(t) \left(1 - \frac{dt}{\tau}\right) + \frac{dt}{\tau} x(t) \quad (5.12)$$

in cui $x_f(0) = x(0)$.

Applicando la (5.12) al segnale del giroscopio di imbardata, si ottiene il grafico in figura 37. Si distinguono i picchi che rappresentano gli otto cambi di direzione, anche se il loro segnale non è comunque estremamente forte rispetto ai possibili rumori dei sensori, nonostante in questo caso la curva effettuata sia piuttosto stretta (figura 30). L’applicazione del filtro al segnale del giroscopio di imbardata ha permesso di isolare e evidenziare le curve durante la camminata, che altrimenti sarebbero rimaste nascoste e non visibili nell’andamento originario. Inoltre, è possibile stabilire anche il verso della rotazione: se il picco è positivo la rotazione è avvenuta in senso antiorario (guardando il piede dall’alto), mentre se è negativo la rotazione è avvenuta in senso orario.

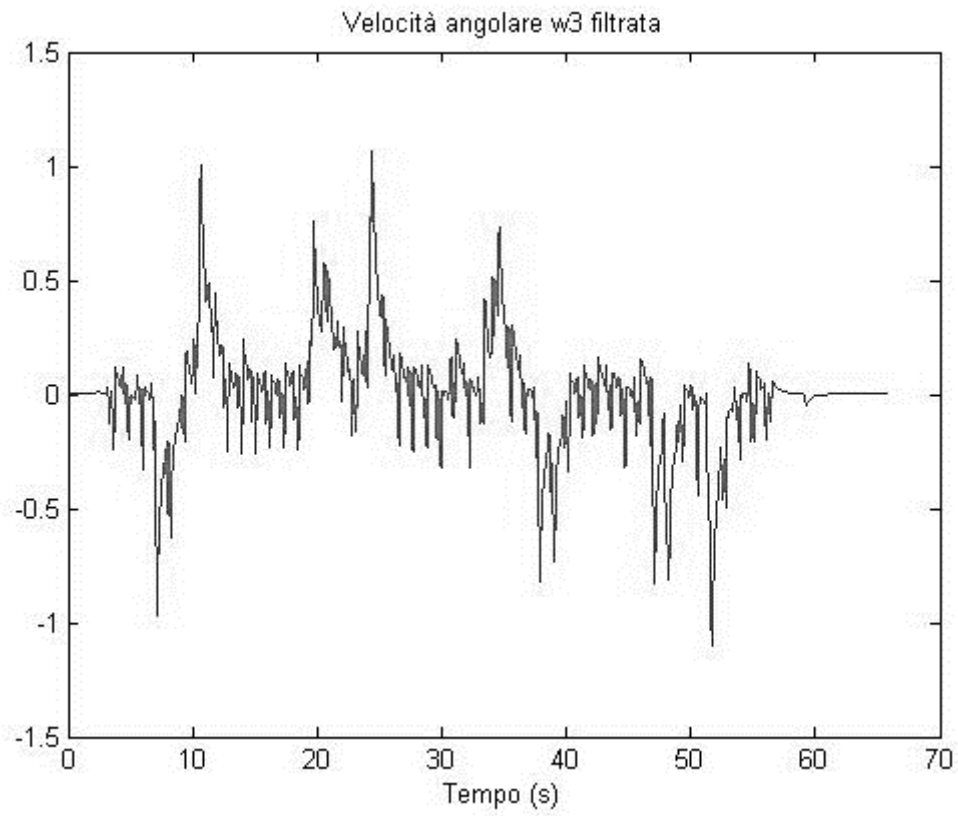


Figura 37: Segnale del giroscopio di imbardata filtrato

Conclusioni

Nel corso della tesi è stato esposto il metodo con cui è possibile ottenere risultati soddisfacenti nell'ambito della "pedestrian navigation", ovvero riuscire a stabilire il percorso effettuato a piedi in assenza di un segnale GPS o di altro tipo, contando dunque solamente sui sensori inerziali e sull'integrazione dei dati che si ottengono da essi. Innanzitutto, è stato progettato un modello matematico per simulare la camminata e l'ottenimento di dati IMU per poi provare con l'integrazione a calcolare la distanza percorsa; questo metodo è stato successivamente applicato a dati ottenuti dagli accelerometri presenti in un comune smartphone, senza ottenere il successo desiderato. Ciò ha portato alla ricerca di un algoritmo che possa compensare la serie di disturbi e problematiche legate ai sensori inerziali, e tramite il progetto OpenShoe è stata individuata una soluzione dagli ottimi risultati, da cui è stato poi dedotto il modello teorico su cui si è basato il codice MATLAB messo a disposizione; questo in particolare sfrutta il rilevamento degli istanti in cui il piede risulta stazionario sulla base dei valori indicati dagli accelerometri e dai giroscopi, per poi correggere il vettore dello stato tramite il filtro di Kalman. Poiché lo studio e la verifica di questo algoritmo sono stati portati avanti con dati messi a disposizione sempre online dal progetto già citato, uno sviluppo futuro potrebbe prevedere la raccolta autonoma di dati da un'altra piattaforma IMU per corroborare la validità del codice, con un'elaborazione dei dati in post-processing come presentato in questa tesi, effettuando prove sia su circuiti chiusi che aperti. Un ulteriore passo potrebbe inoltre condurre alla realizzazione di un sistema che possa calcolare in tempo reale la posizione del piede sfruttando lo stesso algoritmo, affiancando dunque alla piattaforma IMU un processore e trovando il sistema per comunicare con esso a distanza, seguendone i calcoli e risultati.

L'ultima parte della tesi si è invece concentrata sull'analisi in frequenza dei dati dai sensori inerziali messi a disposizione dal progetto OpenShoe, da cui è stata dedotta la possibilità di rilevare la presenza di un cambio di direzione, precisandone anche il verso, studiando il segnale in particolare del giroscopio verticale (di imbardata) alle basse frequenze. In sviluppi futuri occorrerà provare a stimare non solo il cambiamento di direzione, ma anche l'entità della virata, in base alla grandezza dell'accelerazione trasversale misurata. Infine, poiché nel percorso analizzato venivano effettuati cambi di direzione piuttosto stretti, sarebbe interessante ottenere dati nel caso in cui le curve siano più "larghe", e verificare che anche in questo caso è possibile distinguere dal segnale del giroscopio il cambio di direzione.

Riferimenti

- [1] M. Zanzi “Dispense di Elaborazione Dati per la Navigazione – Accelerometri e Gyros”, 01/04/2015 [online]
- [2] G. Sartor “Progetto di un sistema di navigazione al coperto”, 18/02/2010 [online] pp. 20-21
- [3] M. Zanzi “Dispense di Elaborazione Dati per la Navigazione – Accelerometri e Gyros”, pp. 11-12 [online]
- [4] Dynamic Walking MATLAB Simulation Guide [Online], Disponibile al sito http://ruina.tam.cornell.edu/research/topics/locomotion_and_robotics/ranger/ranger_paper/Reports/Ranger_Robot/control/simulator/simplestwalker.html
- [5] Mathworks, “Simulink – simulazione e progettazione model-based” [online] Disponibile al sito <https://it.mathworks.com/products/simulink.html>
- [6] M. Zanzi “Dispense di controlli automatici – Equazioni e trasformate di Laplace” a.s. 2015-2016 [online] pp. 3-4
- [7] M. Zanzi “Dispense di controlli automatici – Antitrasformate e sistemi del I e II ordine” a.s. 2015-2016 [Online] p. 16
- [8] K. Xu, J. Wei, B. Zhang, W. Yang “A Robust Method to detect Zero Velocity for improved 3D Personal Navigation using Inertial Sensor” [Online]
- [9] Rotoview “Sensor Kinetics App” [Online] www.rotoview.com/sensor_kinetics.html
- [10] OpenShoe project, www.openshoe.org
- [11] J. Nilsson, I. Skog, P. Händel, K. V. S. Hari “Foot-mounted INS for Everybody – An Open-Source Embedded Implementation” [Online]
- [12] M. Zanzi “Dispense di Elaborazione Dati per la Navigazione – Sistemi di riferimento e assetto” [Online] pp. 3-5
- [13] M. Zanzi “Dispense di Elaborazione Dati per la Navigazione – Sistemi e Stima dello Stato” [Online]
- [14] J. A. Farrell “Aided Navigation”, 2008, pp. 169-171
- [15] J. A. Farrell “Aided Navigation”, 2008, pp. 172-173

- [16] I. Skog, J. Nilsson, P. Händel “Evaluation of Zero Velocity Detectors for Foot-Mounted Inertial Navigation System” [Online]
- [17] B. Bona “Quaternioni” DAUIN – Politecnico di Torino, 2008 [Online]
- [18] M. J. Baker “Maths – Conversion Matrix to quaternion” Euclidean Space webiste [Online]
- [19] J. Nilsson, I. Skog, P. Händel, K. P. S. Hari “Foot-mounted INS for Everybody – An Open- Source Embedded Implementation” p. 5 [Online]
- [20] “Analisi in frequenze di segnali campionati”, dispense di Misure Elettroniche, Università di Cagliari, 2010 [Online]
- [21] V. Liberali “Teoria dei Segnali – Densità Spettrale di energia e di potenza; Campionamento e teorema di Shannon” 08/11/10, Università degli Studi di Milano [Online]
- [22] “Periodogram” www.mathworks.com [Online]