



UNIVERSITÀ POLITECNICA DELLE MARCHE

FACOLTÀ DI INGEGNERIA

Corso di Laurea in Ingegneria Biomedica

**Studio di metodi per il conteggio dei
passi da segnali accelerometrici**

**Analysis of methods for steps counting
from accelerometer signals**

Tesi di laurea di:

NOEMI ADA FRONZI

Relatore: Chiar.ma

Prof.ssa SUSANNA SPINSANTE

Correlatore: Chiar.ma

Prof.ssa STEFANIA CECCHI

Anno Accademico 2019-2020

Abstract

La presente tesi di laurea descrive l'analisi e i risultati ottenuti da metodi per il conteggio dei passi attraverso segnali accelerometrici generati da un accelerometro posto su una calzatura, nella zona esterna e posteriore del tallone. L'obiettivo finale è quello di poter implementare un algoritmo che permetta di ottenere il numero di passi effettuati, confrontandoli con quelli registrati tramite un'app esterna assunta come termine di riferimento.

Dall'analisi dei vari studi proposti in ambito scientifico, si passa all'acquisizione ed elaborazione dei dati registrati con l'utilizzo della *smartshoe*, fino al confronto di due metodi messi a punto per la stesura degli algoritmi.

I risultati ottenuti mostrano come il procedimento scelto, eseguito utilizzando un metodo a soglia che intercetta le intersezioni della retta relativa al valore medio del segnale accelerometrico con il segnale stesso, sia più impreciso rispetto a quello implementato utilizzando una funzione *findpeaks* di identificazione dei picchi del segnale stesso.

Il lavoro si conclude con la proposta di un metodo alternativo, o complementare, di segnali di pressione per il conteggio dei passi tramite *smartshoe*.

Indice

INTRODUZIONE	1
CAPITOLO 1: METODI E ALGORITMI PER IL CONTEGGIO DEI PASSI	3
1.1 Classificazione e analisi tramite confronto tra i metodi	3
1.2 Considerazioni sui metodi analizzati	14
CAPITOLO 2: ACQUISIZIONE DEI SEGNALI DELLA SMARTSHOE	15
CAPITOLO 3: ELABORAZIONE DEI SEGNALI ACCELEROMETRICI IN MATLAB	18
3.1 Composizione del vettore Ampiezza Accelerazione	18
3.2 Metodo rilevamento picchi con retta mean	21
3.3 Metodo rilevamento picchi con Findpeaks	25
3.4 Confronto tra i due metodi	31
CAPITOLO 4: ELABORAZIONE DEI SEGNALI DI PRESSIONE IN MATLAB	32
4.1 Acquisizione delle componenti del segnale di pressione	32
4.2 Composizione del segnale di Pressione Media	33
CONCLUSIONI	36
BIBLIOGRAFIA	38
RINGRAZIAMENTI	40

Introduzione

Negli ultimi anni il mercato dei dispositivi indossabili, con diverse applicazioni che spaziano dall'entertainment, al controllo dell'attività fisica, al monitoraggio biomedico, è sempre più in rapida crescita.

Per questo motivo, si è sentita sempre più la necessità di sviluppare algoritmi di monitoraggio dell'attività fisica più precisi e affidabili, cercando di individuare e testare metodi che potessero fornire dati attendibili.

L'oggetto di questa tesi di laurea, frutto del lavoro svolto durante il tirocinio, è la stesura di un algoritmo che permetta di effettuare il conteggio dei passi tramite una calzatura sensorizzata che chiameremo *smartshoe*. La calzatura è dotata di tre sensori e di un accelerometro connessi ad un'app tramite un dispositivo Bluetooth integrato nella calzatura, che consente di raccogliere il segnale accelerometrico e i segnali di forza generati durante le varie acquisizioni.

La prima parte dell'elaborato si occupa di analizzare i vari metodi implementati e descritti in pubblicazioni scientifiche, in modo da valutare la scelta operativa che sarà poi utilizzata per l'algoritmo.

Successivamente, si passa all'acquisizione dei dati tramite l'uso della calzatura, indossata in quattro diverse situazioni (camminata su un piano, su un tapis roulant, salita e discesa di una scalinata) e alla loro elaborazione in ambiente software Matlab.

Infine, si confrontano i risultati ottenuti con il numero di passi contati da un'app scaricata sullo smartphone, attivata durante le acquisizioni effettuate indossando la *smartshoe*.

L'obiettivo finale è quello di valutare quale dei metodi scelti e poi implementati sia più adatto per questo tipo di dispositivo indossabile e per il suo eventuale uso come contapassi.

CAPITOLO 1

Metodi e algoritmi per il conteggio dei passi

1.1 Classificazione e analisi tramite confronto tra i metodi.

Negli ultimi anni, il mercato dei dispositivi indossabili è cresciuto esponenzialmente, rendendo sempre più accattivante l'idea di poter realizzare algoritmi che permettano un monitoraggio dell'attività fisica delle persone (Fig.1) sempre più accurato e affidabile.

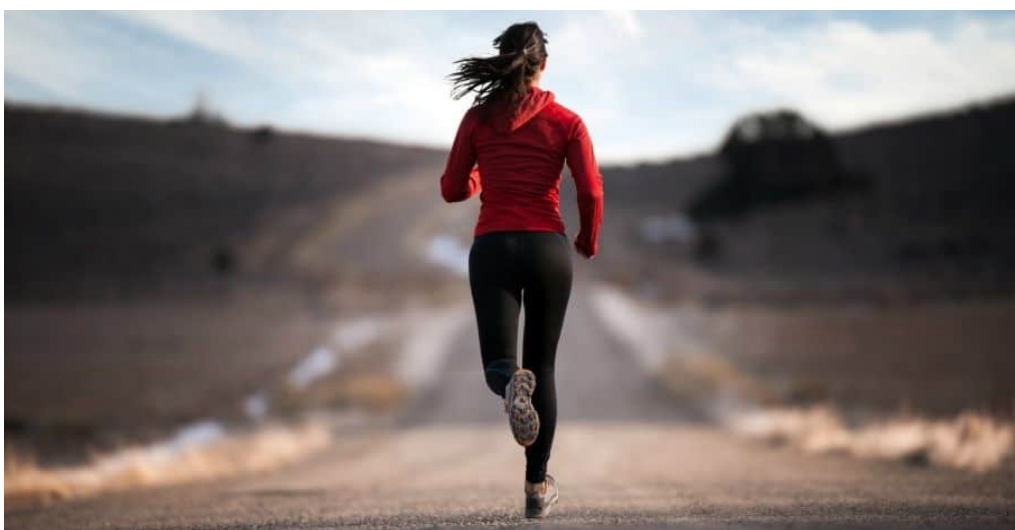


Fig. 1 I contapassi sono comunemente utilizzati dagli sportivi per il monitoraggio dell'attività fisica.

Di particolare interesse è il conteggio dei passi, che può essere effettuato sia tramite smartphone, quello più comunemente usato, che attraverso smartwatch o smartshoe, mostrati in Fig. 2.



Fig.2 Smartwatch e smartshoe

Nel corso degli anni vi sono stati numerosi tentativi di sviluppare algoritmi di conteggio dei passi basati su un segnale accelerometrico.

In generale, gli studi esistenti possono essere classificati nei seguenti approcci: *dominio del tempo*, *dominio della frequenza* e *raggruppamento delle caratteristiche*.

Gli approcci nel *dominio del tempo* includono:

- Metodi basati sull'uso di una **soglia** che contano le fasi valutando se i dati sensoriali soddisfano alcuni valori predefiniti. Ad esempio, in ^[1] l'algoritmo progettato si basa sull'analisi della soglia dell'accelerazione. In particolare, dato che il limite superiore dell'accelerazione è sempre diverso, in quanto varia periodicamente, dopo aver identificato un valore massimo, i dati al 90% dell'accelerazione di quel valore vengono considerati come passi. Quindi, se i campioni del segnale di accelerazione raggiungono o superano la soglia in ciascun ciclo, il sistema trasmette un nuovo passo.
- Il **rilevamento dei picchi** stima le fasi del passo in base al numero di picchi dati da una sequenza di dati sensoriali, come mostrato in ^[2] ed è comunemente utilizzato con la soglia, per cui il valore di accelerazione dei picchi viene verificato se supera una soglia predefinita. Un esempio è presentato in ^[3], dove gli autori implementano un algoritmo in cui viene valutato il valore massimo all'interno di un range, che viene assunto come soglia per il rilevamento del picco. Quindi, se un campione nel range considerato presenta un valore

maggiore della media valutata, viene identificato un picco e considerato come passo.

- Il conteggio dei **passaggi per lo zero (ZUPT)** conta i passi rilevando il numero di valori nulli nei dati sensoriali. Sia l'approccio di rilevamento dei picchi che l'approccio di attraversamento dello zero cercano i periodi inerenti alla natura ciclica della deambulazione, usando l'entità delle accelerazioni o delle velocità angolari.

In ^[4] questo approccio viene utilizzato per caratterizzare il ciclo di camminata, dove la velocità cambia a causa di accelerazioni e decelerazioni, diventando zero periodicamente. In particolare, l'intervallo di accelerazione corrisponde al passaggio dall'elevazione del tallone all'oscillazione del piede nel punto più alto. L'intervallo di decelerazione corrisponde invece al passaggio dal punto più alto al momento in cui il piede è piatto sul pavimento. Quello in cui la velocità è zero è il passaggio dal punto in cui il piede è piatto al sollevamento del tallone.

Tramite il rilevamento degli intervalli di velocità a zero, gli autori costruiscono un algoritmo per il conteggio dei passi, in cui ogni intervallo contiene le tre diverse variazioni della velocità che corrispondono proprio ad un passo.

- L'ultimo approccio utilizzato nel dominio del tempo è quello dell'**autocorrelazione**, che rileva i periodi ciclici direttamente nel dominio del tempo attraverso la valutazione dell'autocorrelazione. Come spiegato in ^[5], l'intuizione di questo metodo sta nel fatto che la camminata dell'uomo ha una natura ciclica, in cui l'alternarsi del piede sinistro e destro può essere considerato un periodo. Questo implica che, se l'utente cammina, l'autocorrelazione aumenterà con la periodicità della camminata dell'utente e, grazie alla sua stima, si effettua il conteggio passi.

Per quanto riguarda invece gli approcci nel *dominio della frequenza*, essi si concentrano sul contenuto in frequenza di intervalli di misurazioni effettuate e si basano sulla trasformata di Fourier:

- **STFT (Short Time Fourier Transform)** utilizzata in ^[6] per dividere il segnale in intervalli e per calcolare un numero frazionario di passi per ogni intervallo, dividendo la larghezza di questo per il periodo di camminata, valori poi sommati per stimare i passi effettuati;
- **FFT (Fast Fourier Transform)**, utilizzata in ^[7] e ^[8] per trasformare i segnali dal dominio del tempo a quello della frequenza, da cui poi si ricavano i segnali periodici e si applica l'algoritmo per il conteggio dei passi.

Infine, gli approcci per *raggruppamento di funzionalità* utilizzano algoritmi di apprendimento automatico, ad esempio gli **Hidden Markov Models (HMMs)** o l'algoritmo **KMeans**, per classificare le attività in base al dominio del tempo e a funzioni del dominio della frequenza estratte da dati sensoriali. Come illustrato in ^[6], i due metodi vengono utilizzati sia per il rilevamento dell'andatura, quindi camminata, corsa, ecc, che per il conteggio dei passi. In particolare gli HMMs, quando applicati alla porzione di dati che rappresentano la camminata, vengono utilizzati per distinguere le diverse fasi dell'andatura, ovvero tallone-punta-tallone-piede fermo, assegnando uno stato per ogni picco positivo del segnale e un altro stato per ogni picco negativo del periodo del passo.

Invece, per l'approccio con l'algoritmo KMeans, che ha l'obiettivo di minimizzare la varianza totale all'interno di un gruppo, vengono partizionati i vettori di funzioni in finestre temporali scorrevoli in due clusters (picchi positivi e negativi) usando *l'algoritmo di Lloyd*, il quale assegna i punti di ingresso a ogni partizione, quindi calcola il centroide di ogni gruppo.

Come dimostrano tutti gli studi fatti negli anni, l'accuratezza del conteggio dei passi è influenzata da molteplici fattori, come il posizionamento dei dispositivi, le abitudini individuali di camminata, ma anche il tipo di sensore, come spiegato in ^[9]. Gli autori, infatti, studiando un dispositivo da polso, sottolineano il fatto che, essendo caratterizzati da una bassa potenza e affetti da errore, gli algoritmi progettati per il conteggio dei passi devono essere di bassa complessità e produrre risultati accurati per le varie modalità di trasporto e per le variazioni di velocità. Per questo, utilizzano il metodo del rilevamento

picchi, scartando algoritmi più complessi che possono non essere adatti per questo tipo di dispositivi.

Inoltre, bisogna tener conto anche del fatto che la velocità di camminata è influenzata da vari fattori che possono essere emotivi (quando si è agitati si tende a camminare più velocemente) oppure ambientali (in una strada affollata siamo costretti a rallentare).

Sebbene gli algoritmi di conteggio dei passi esistenti siano in grado di affrontare una vasta gamma di velocità di camminata, le situazioni in cui si presentano variazioni di velocità sono raramente prese in considerazione, portando a risultati non precisi.

Ad esempio, poiché le diverse velocità di camminata generalmente comportano cambiamenti significativi nell'ampiezza e nella frequenza delle misurazioni inerziali, come accelerazioni e velocità angolari, è difficile distinguere picchi e interferenze reali, andando quindi a influire sulla precisione dell'algoritmo. Questo problema è stato ampiamente discusso in ^[10], dove gli autori hanno scartato i metodi nel dominio del tempo, in quanto la periodicità della deambulazione non può essere facilmente distinta poiché la somiglianza tra due passi vicini risulta essere molto ridotta. Per questo, si decide di calcolare con precisione l'area determinata dalle curve delle frequenze di camminata, denominato metodo ISC, oppure di valutare la lunghezza del passo in base alle frequenze di camminata, chiamato metodo ADSC.

In ^[1], dato che l'accelerazione varia a seconda del ritmo di camminata, dopo aver effettuato un'analisi della soglia dell'accelerazione, identificata come il confine di un sistema di dominio numerico, e ottenuto il valore di picco massimo, i dati al 90% dell'accelerazione di picco vengono presi come valore numerico e considerati come passi.

Il diagramma di flusso in Fig.3 descrive i passaggi effettuati per la progettazione dell'algoritmo in ^[1].

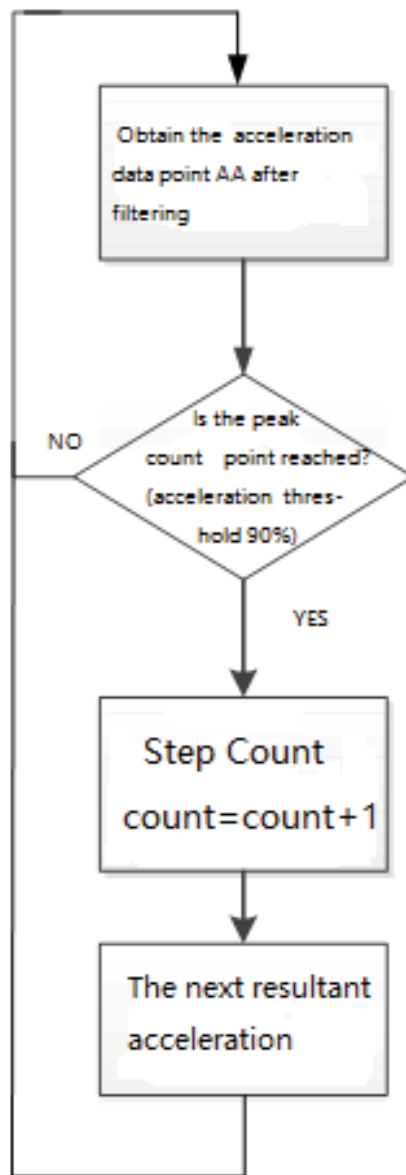


Fig. 3 Diagramma di flusso dell'analisi delle soglie e dell'algoritmo contapassi di [1].

La limitazione del metodo che utilizza un valore di soglia sta proprio nel fatto che è quasi impossibile trovare una soglia ottimale per tutti i casi, specialmente quando si tratta di smartphone, in cui bisogna tenere conto di una molteplicità di casi maggiore (quando è in tasca, quando si risponde a una telefonata, ecc), anche se risulta essere l'approccio più semplice.

Anche l'accuratezza dell'algoritmo di conteggio dei passi basato sul metodo di autocorrelazione risulta essere bassa, a causa del fatto che la somiglianza tra le forme d'onda adiacenti diminuisce per le diverse velocità di camminata. Per ovviare a questo problema, gli autori di ^[11] utilizzano la componente orizzontale dell'accelerazione e la componente verticale della velocità angolare per valutare le autocorrelazioni e mostrare una buona accuratezza di rilevamento, ma soffrono di elevati costi di calcolo per la trasformazione dei sistemi di riferimento.

Invece, i metodi di rilevamento picchi e quello dell'attraversamento dello zero cercano i periodi inerenti alla natura ciclica della deambulazione usando l'entità delle accelerazioni o delle velocità angolari. Questi possono ottenere prestazioni migliori con accelerazioni verticali, ma sono imprecisi se i sensori non sono saldamente attaccati a parti del corpo.

In ^[12], per rilevare i passi effettivi, si identifica il picco tramite la differenza tra la cresta dell'onda adiacente e la depressione dell'onda che descrive l'accelerazione. I risultati che risultano essere al di sotto di un certo valore considerato come valore massimo, che può essere raggiunto dall'accelerazione mentre si effettua un passo, vengono contati, mentre gli altri vengono esclusi. A questo punto, si calcola il massimo dell'accelerazione tra i due punti a zero adiacenti, in cui l'accelerazione è zero e la velocità è massima, come mostrato in Fig.4.

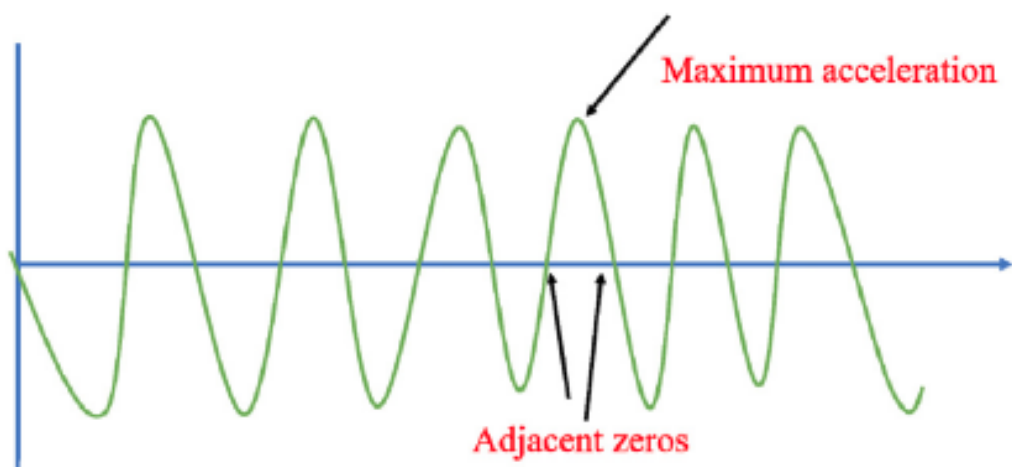


Fig.4 Metodo rilevamento passi in ^[12].

I metodi che sfruttano il dominio della frequenza e gli approcci per funzionalità, invece, vengono utilizzati per migliorare l'accuratezza degli algoritmi in cui si utilizzano i metodi nel dominio del tempo.

In ^[8], l'algoritmo sviluppato si basa sulla soglia dei dati raccolti dall'accelerometro, ma, anziché applicarlo direttamente ai dati ottenuti dal sensore (Fig.5), si applica la trasformata di Fourier, come mostrato in Fig.6 e Fig.7, la quale converte il segnale da una rappresentazione nel dominio del tempo a quella nel dominio della frequenza. Successivamente, l'algoritmo rileva i segnali periodici e li divide in componenti armoniche, a cui poi si applicano le soglie.

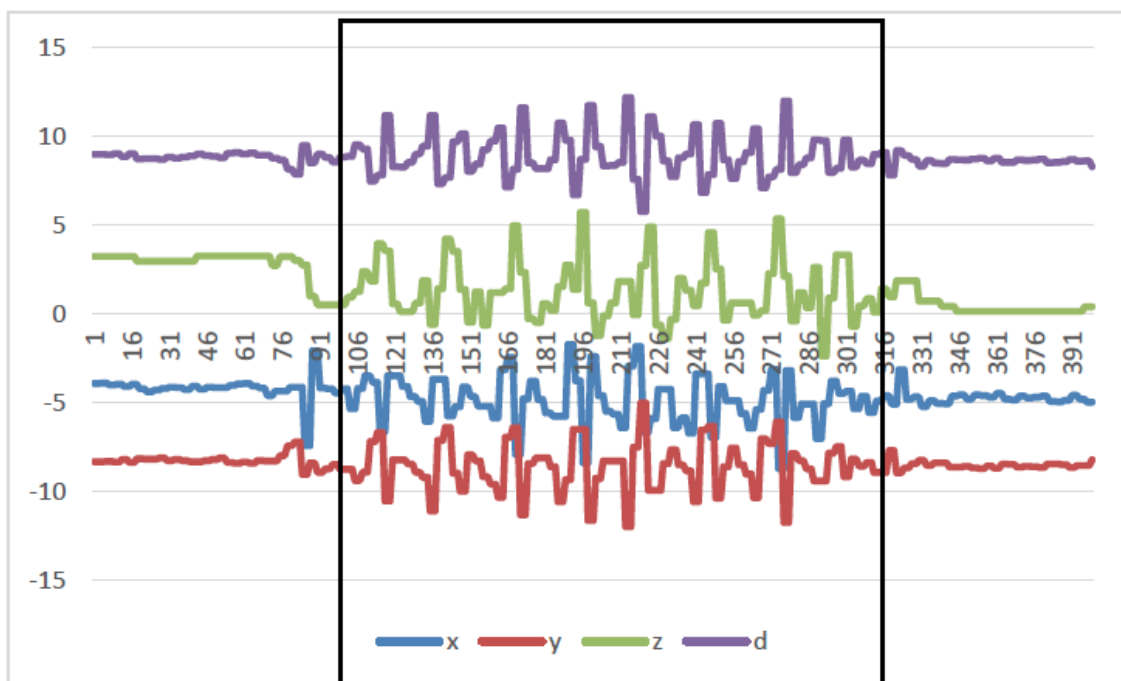


Fig. 5 Segnale accelerazione per lo stare in piedi, per la camminata e per lo stare seduti rilevato in ^[8].

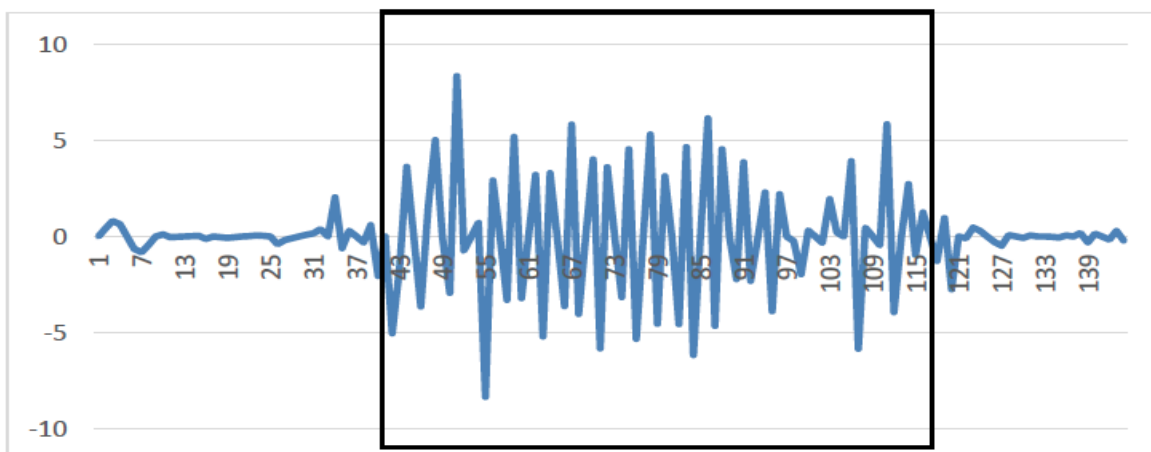


Fig. 6 Parte immaginaria dei dati convertiti con la FFT per lo stare in piedi, per la camminata e per lo stare seduti rilevato in ^[8].

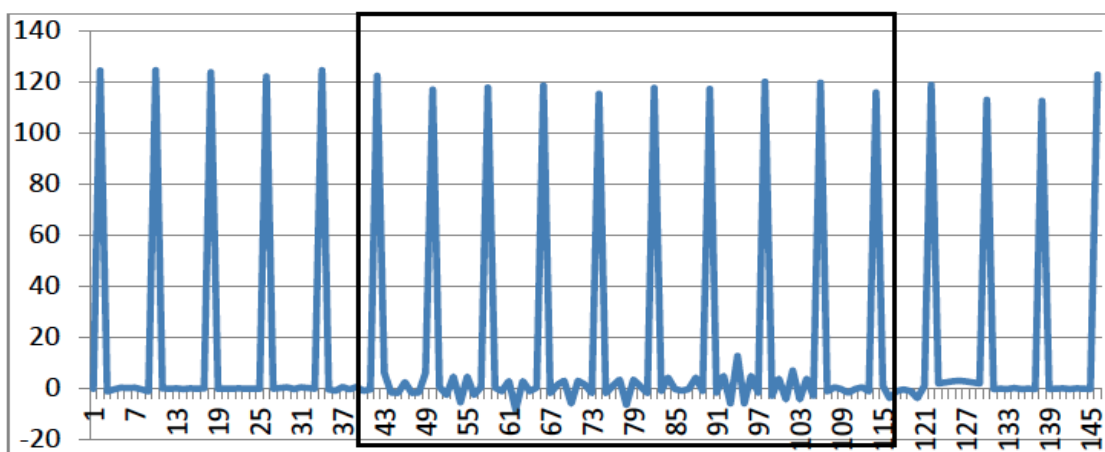


Fig. 7 Parte reale dei dati convertiti con la FFT per lo stare in piedi, per la camminata e lo stare seduti rilevato in ^[8].

Lo stesso accade in ^[4], dove l'algoritmo per il rilevamento passi si basa sul metodo ZUPT, in cui si tenta di rilevare gli intervalli dove la velocità risulta essere zero, che corrisponde al punto in cui il piede è piatto sul terreno. Per il rilevamento della velocità nulla, si utilizza l'Hidden Markov Model, in cui i cicli di andatura vengono modellati come una macchina a stati finiti (*FSM*, dall'inglese *Finite State Machine*) e ogni ciclo di andatura è diviso in tre stati: intervalli a velocità zero, intervalli di accelerazione e intervalli di decelerazione, come mostra il diagramma in Fig.8. Si determinano poi le pendenze delle velocità nei vari intervalli per capire quando si passa da uno stato all'altro.

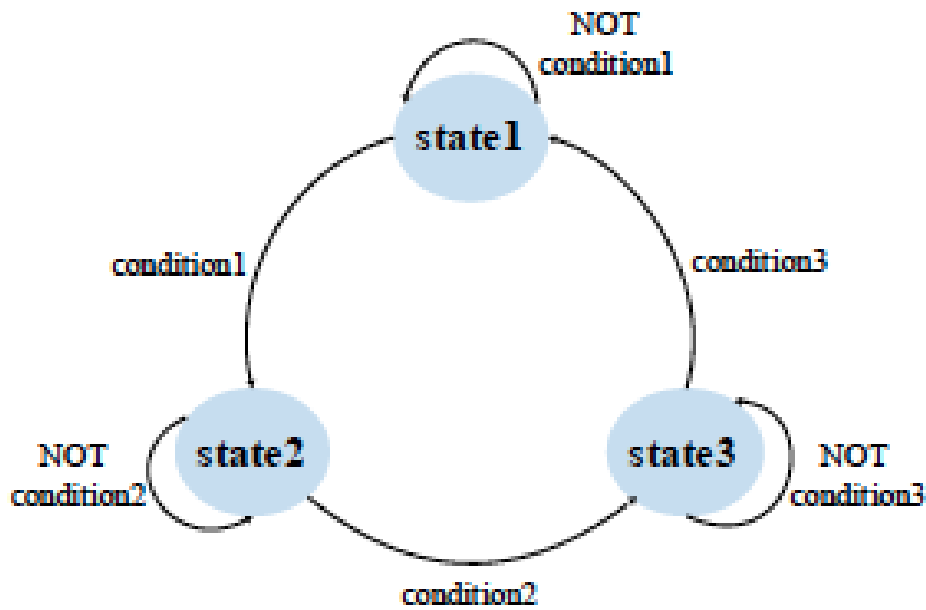


Fig. 8 Diagramma di transizione di stato proposto in ^[4].

Entrambi i metodi riescono a migliorare gli algoritmi che sfruttano gli approcci nel dominio del tempo, ma il problema che li accomuna sta nel fatto che risultano avere costi computazionali elevati, come è possibile vedere in ^[13].

Un approccio completamente differente è quello adottato in ^[14], in cui viene presentato un metodo di monitoraggio dei passi attraverso l'utilizzo di una smartshoe basata sulla comunicazione wireless che utilizza il modulo di trasmissione Wi-Fi con resistori sensibili alla forza (FSR), distribuiti sotto la soletta, confrontato poi con un' app contapassi di uno smartphone.

Inizialmente è stato calcolato il valore di Pressione media (P_{avg}) facendo la media tra i valori di pressione registrati dai vari sensori. Per l'algoritmo del conteggio passi, invece, si è introdotto il valore di ΔP_{avg} , utilizzando α e β come soglie per la rilevazione dei picchi, come mostrato in Fig.9.

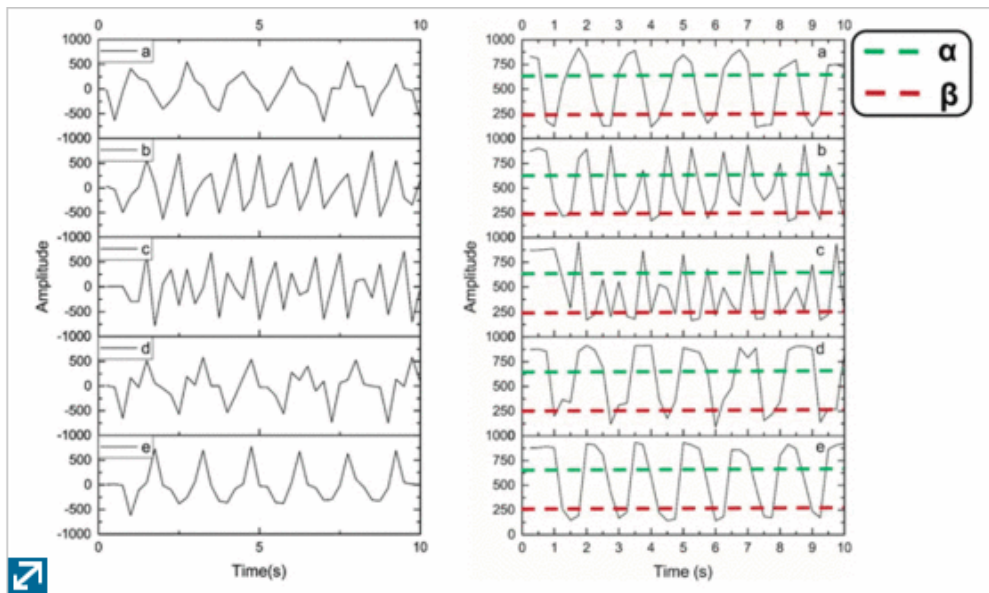


Fig. 9 Segnali di Pressione rilevati in ^[14]: A sinistra: P_{avg} nella camminata (a), nel jogging (b), nello sprint (c), nella salita delle scale (d), nello scendere le scale (e). A destra: ΔP_{avg} nella camminata (a), nel jogging (b), nello sprint (c), nella salita delle scale (d), nello scendere le scale (e).

Inoltre, il valore di trigger viene utilizzato per tracciare brevi stati del ciclo dell'andatura, comprese le fasi di posizione e di oscillazione. In altre parole, i passaggi possono essere approssimativamente conteggiati come il numero di passi.

Calcolando l'accuratezza in percentuale del conteggio passi effettuato, gli autori sono riusciti a dimostrare che la smartshoe fornisce una precisione nettamente superiore rispetto a quella dello smartphone.

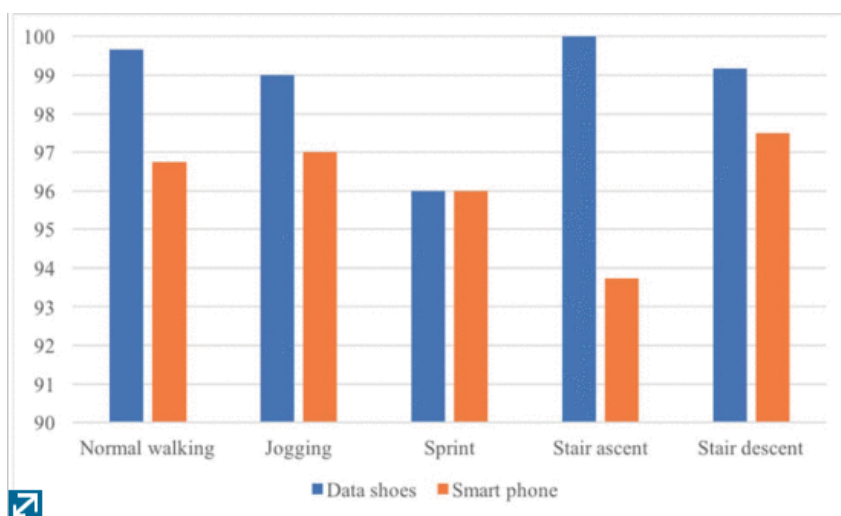


Fig. 10 Confronto di precisione tra smartshoe e smartphone in ^[14].

1.2 *Considerazioni sui metodi analizzati*

Analizzando gli studi sviluppati negli anni da diversi autori e relativi ai metodi per il conteggio dei passi, si è potuto notare come non ci sia un modo più preciso di un altro, tenendo conto del fatto che le velocità di camminata sono influenzate da dinamiche dovute a emozioni individuali, abitudini di camminata o influenze ambientali, ma anche dalla diversità dei dispositivi utilizzati, che possono essere posizionati sulla pianta del piede, in vita, sul polso, sulle mani, ecc.

Generalmente però, gli approcci più utilizzati sono quelli basati sulla rilevazione della soglia e dei picchi, in quanto più semplici da utilizzare e che comportano anche una minore spesa computazionale, con il limite però di non poter assegnare una soglia universale, a causa dei diversi valori che la velocità può assumere.

Al contrario, gli approcci basati sul dominio della frequenza, che risultano essere intuitivamente più adatti in quanto la camminata presenta un andamento ciclico, possono fornire un'accuratezza maggiore, a spese della semplicità dell'algoritmo, che tende ad essere più complesso.

Come ultimo metodo analizzato, si è scelto un approccio diverso basato sui sensori resistivi FSR, sfruttando la pressione anziché l'accelerazione, con il risultato di essere più preciso ma che, ovviamente, non può essere applicato a qualsiasi dispositivo deputato al conteggio passi.

CAPITOLO 2

Acquisizione dei segnali della Smartshoe

Il lavoro svolto durante il tirocinio riguarda la progettazione di un algoritmo che permetta di effettuare il conteggio dei passi tramite una smartshoe.

La calzatura è costituita da quattro sensori: tre FSR (Force Sensing Resistor) posti sulla pianta della scarpa che registrano la pressione esercitata dal piede durante la camminata e un accelerometro posto sulla parte esterna della calzatura, come mostrato in Fig. 11,12 e 13. Tutti i sensori sono connessi ad una app fornita dal costruttore stesso, tramite un dispositivo Bluetooth integrato nella calzatura, che consente di raccogliere il segnale accelerometrico e i segnali di forza generati durante le varie acquisizioni.

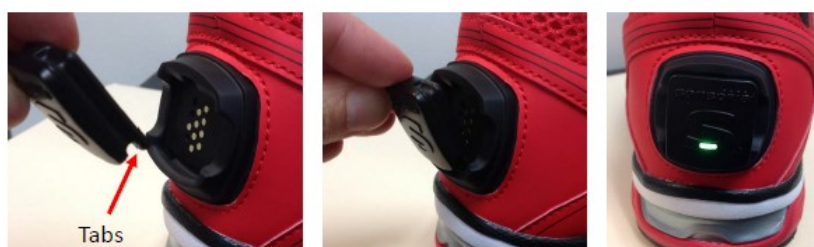


Fig. 11. Accelerometro esterno. Si installa immediatamente dietro alla conchiglia e si illumina subito dopo aver collegato il telefono.

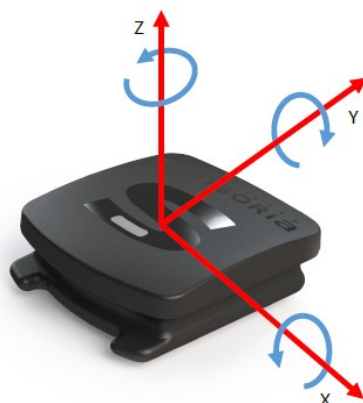


Fig.12 L'accelerometro è un sensore di accelerazione triassiale (assi X,Y e Z), lungo gli assi in cui viene registrato il movimento.

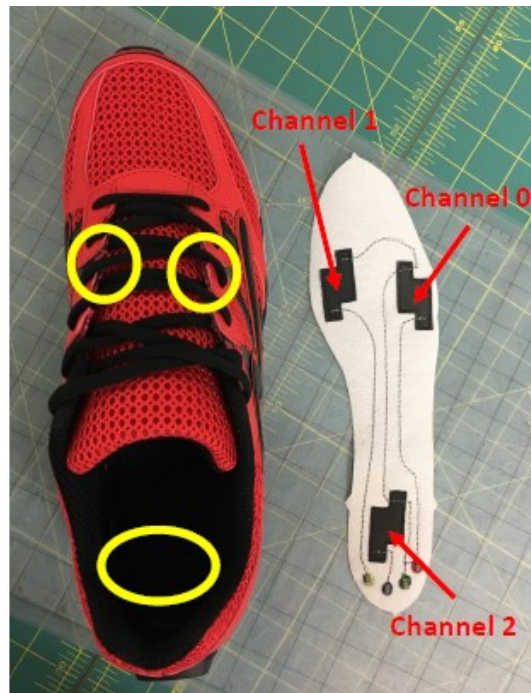


Fig.13 I tre sensori sono posizionati sulla pianta della scarpa, due lungo l'arco trasverso e il terzo su tallone.

Il progetto sviluppato prevede l'analisi dei segnali e la predisposizione di un algoritmo in Matlab, finalizzato al conteggio dei passi compiuti. Il lavoro si basa principalmente su un metodo esposto da ^[15].

Il confronto con il risultato fornito da una app contapassi, installata sullo smartphone e attivata durante le acquisizioni svolte indossando la smart shoe, permette di verificare la validità dell'algoritmo implementato, in seguito alle dovute elaborazioni dei segnali acquisiti.

I segnali vengono acquisiti prendendo in considerazione quattro diverse situazioni: la camminata su una strada rettilinea, la camminata su un tapis roulant, la salita e la discesa delle scale e una quinta acquisizione in cui la calzatura rimane ferma, senza essere indossata.

Viene effettuata con la calzatura indosso una camminata di circa cinque minuti, sia sulla strada rettilinea che sul tapis roulant, mentre per la salita e la discesa si usa una scalinata composta da dieci scalini.

Per verificare la validità dello studio, durante l'acquisizione delle camminate su rettilineo e sul tapis roulant, viene attivata l'app Runtastic Steps Contapassi^[1] installata su Iphone (Fig.14), per registrare il numero di passi effettuato, in modo da confrontarlo successivamente con quello restituito dall'algoritmo.

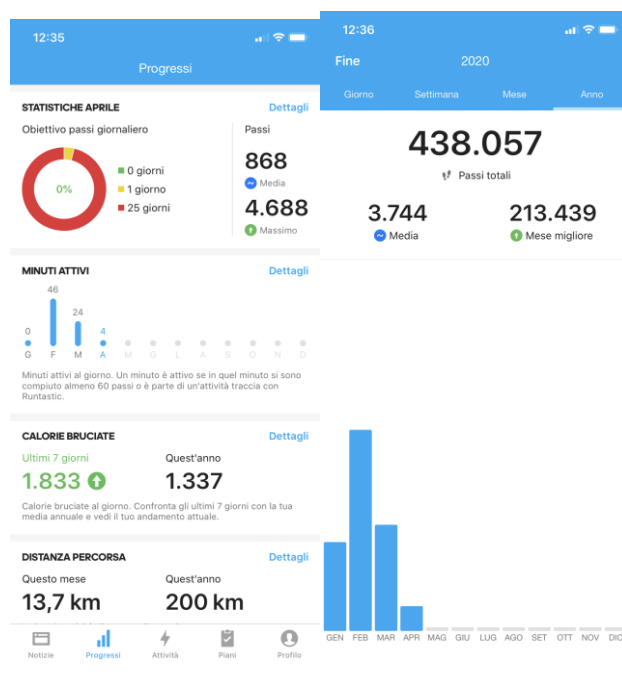


Fig. 14 *Interfaccia della app Runtastic Steps Contapassi*

Tale valore viene preso come riferimento per confrontare il dato ottenuto elaborando i segnali generati dalla smartshoe.

In particolare, per la camminata su strada rettilinea, l'app ha contato 583 passi per cinque minuti di camminata; per il tapis roulant, ha contato 556 passi per cinque minuti di camminata.

I segnali generati dai sensori nella calzatura vengono registrati attraverso l'app fornita dal costruttore, che consente di salvare sotto forma di documento .csv su Excel tutte le grandezze acquisite che descrivono il passo in funzione del tempo.

^[1] <https://www.runtastic.com/blog/it/steps-nuova-app-me-per-contare-i-passi/>

CAPITOLO 3

Elaborazione dei segnali accelerometrici in Matlab

3.1 Composizione del vettore Ampiezza Accelerazione

Dopo aver salvato le varie acquisizioni, si importano tutti i dati in Matlab, in modo da elaborare graficamente i segnali accelerometrici.

Si selezionano dapprima solo le tre componenti dell'accelerazione (lungo x, y e z) come variabili di tipo Column Vector, avendo cura di ripulire ciascuna colonna da eventuali valori NaN, dovuti ad alcune parti di segnale che non sono state acquisite dalla calzatura durante l'effettuazione dei passi.

Viene poi composto il vettore Ampiezza della accelerazione per tutti e quattro i casi acquisiti, utilizzando le tre componenti A_x A_y A_z , calcolandone ogni elemento come la radice quadrata della somma dei quadrati delle corrispondenti componenti:

$$A = (A_x^2 + A_y^2 + A_z^2)^{1/2}.$$

Per eliminare dai segnali l'eventuale errore dovuto al drift dell'accelerometro a bordo della calzatura, si sottrae a tutti i segnali di accelerazione quello prelevato dall'acquisizione a vuoto, in modo da ottenere un risultato più preciso possibile.

Graficando il segnale, nel tentativo di darne una prima valutazione qualitativa, si osserva che, il numero di campioni acquisiti durante una sessione, risulta elevato, avendo il sensore un sample rate di 10 Hz. Per questo motivo, per una visualizzazione più chiara, si sceglie di rappresentare un numero di campioni che va da 1 a 400.

L'algoritmo utilizzato per la costruzione del segnale accelerometrico, nel caso della camminata sul tapis roulant, è il seguente:


```

>> Ax = Ax (1:400); #componente x dell'accelerazione
>> Ay = Ay (1:400); #componente y dell'accelerazione
>> Az = Az (1:400); #componente z dell'accelerazione
>> a = [Ax, Ay, Az];
>> A_tapisroulant = sqrt ((a (:,1)).^2 + (a (:,2)).^2 + (a (:,3)).^2);
>> Ax_vuoto = Ax1 (1:400); #componente x dell'accelerazione acquisita a vuoto
>> Ay_vuoto = Ay1 (1:400); #componente y dell'accelerazione acquisita a vuoto
>> Az_vuoto = Az1 (1:400); #componente z dell'accelerazione acquisita a vuoto
>> a1 = [Ax_vuoto, Ay_vuoto, Az_vuoto];
>> A_vuoto = sqrt ((a1 (:,1)).^2 + (a1 (:,2)).^2 + (a1 (:,3)).^2);
>> A_tapisroulantpulito = A_tapisroulant - A_vuoto; #accelerazione ripulita dal
segnale dell'acquisizione a vuoto
>> plot (A_tapisroulantpulito);

```

Si utilizza lo stesso algoritmo per tutte le altre acquisizioni, ottenendo l'andamento dell'accelerazione.

Figure 15-16-17-18 riportano i grafici ottenuti in tutte le acquisizioni.

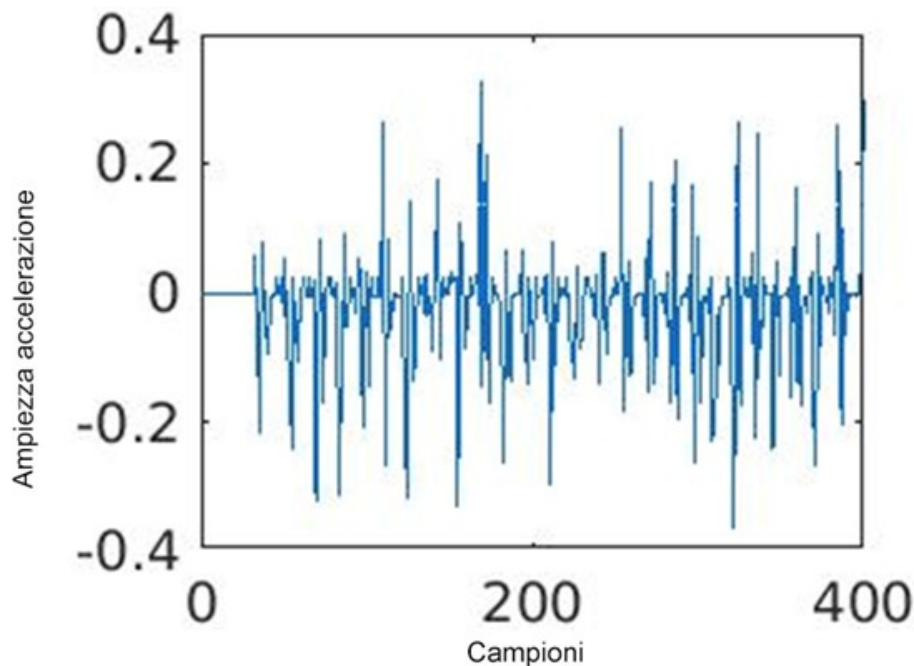


Fig. 15 Segnale accelerometrico camminata su strada rettilinea

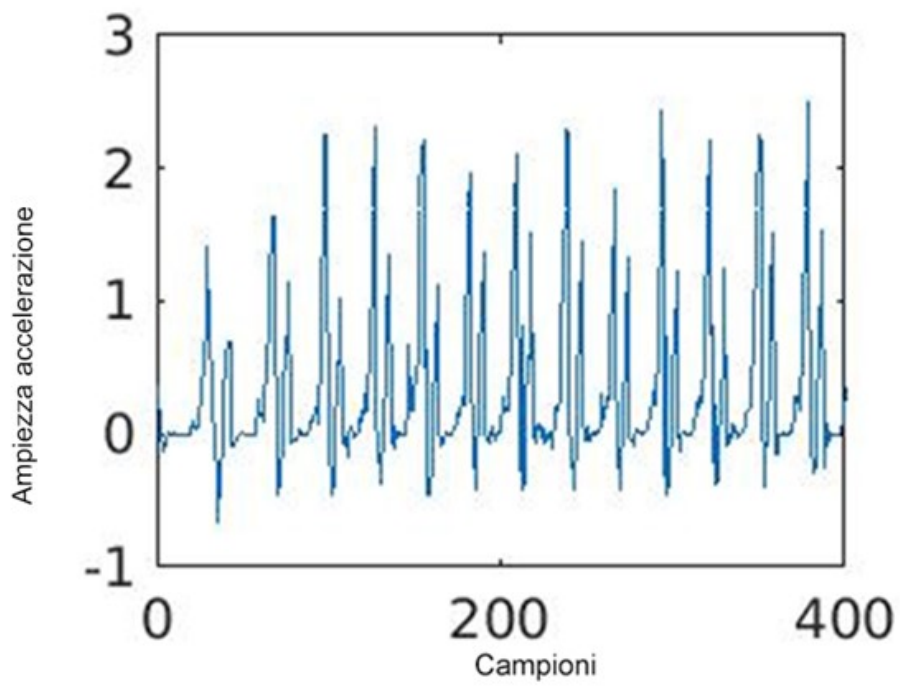


Fig. 16 Segnale accelerometrico su tapis roulant

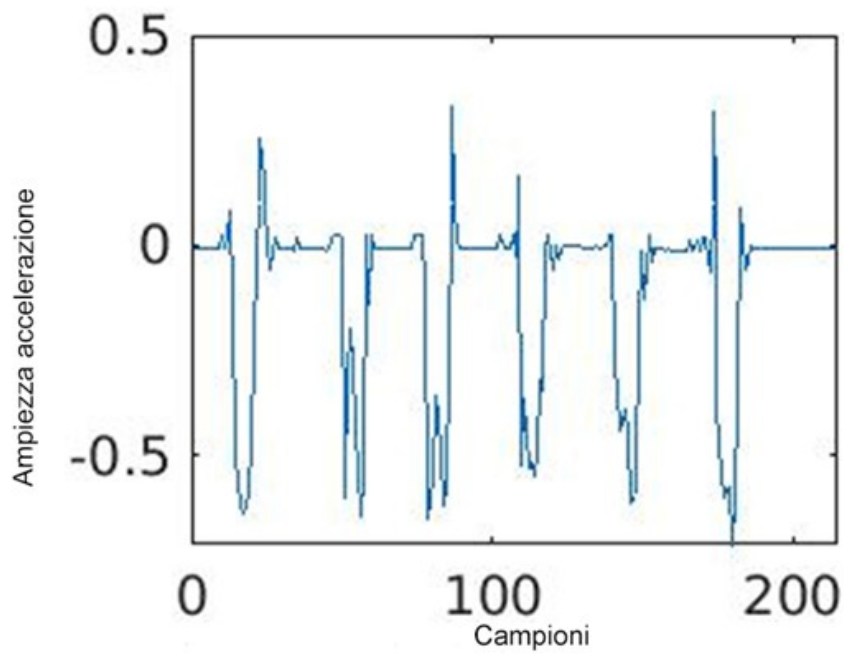


Fig. 17 Segnale accelerometrico salita scale.

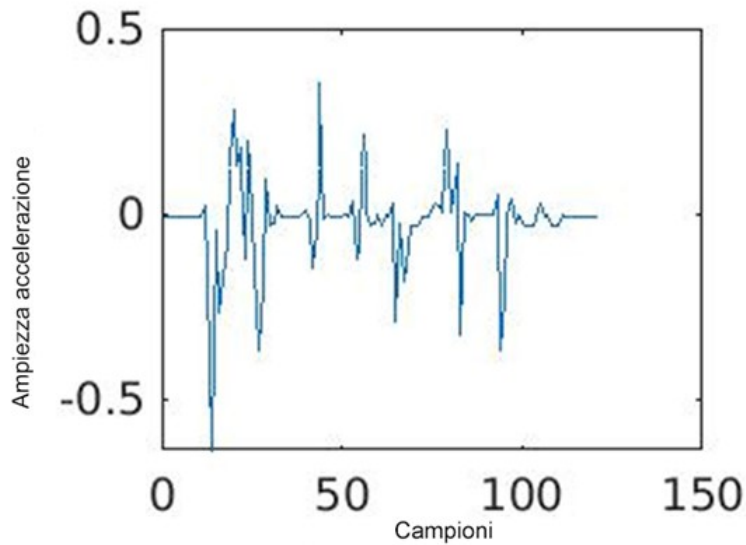


Fig. 18 Segnale accelerometrico discesa scale.

3.2 Metodo rilevamento picchi con retta mean

Il metodo utilizzato per la costruzione dell'algoritmo si basa sulla rilevazione dei picchi del segnale, dove ogni picco rappresenta un passo.

Si individua dapprima una retta che descrive la media del segnale accelerometrico, in modo che ogni due intersezioni di tale retta con i grafici precedentemente ottenuti, possa essere contato un passo.

Si prende in esame il caso della camminata sul tapis roulant, in quanto, essendo un segnale scandito dalla velocità della macchina, risulta essere più lineare e quindi facilita la verifica della validità del metodo usato.

L'algoritmo che si usa per la costruzione della retta e per il successivo confronto con il segnale accelerometrico è il seguente:

```
>>a_mean = zeros (1, length (A_tapisroulant));
>>a_mean (:)= mean (A_tapisroulant);
>>plot (A_tapisroulant);
>>hold on;
>>plot (a_mean);
```

Fig. 19 riporta il grafico ottenuto con il precedente codice.

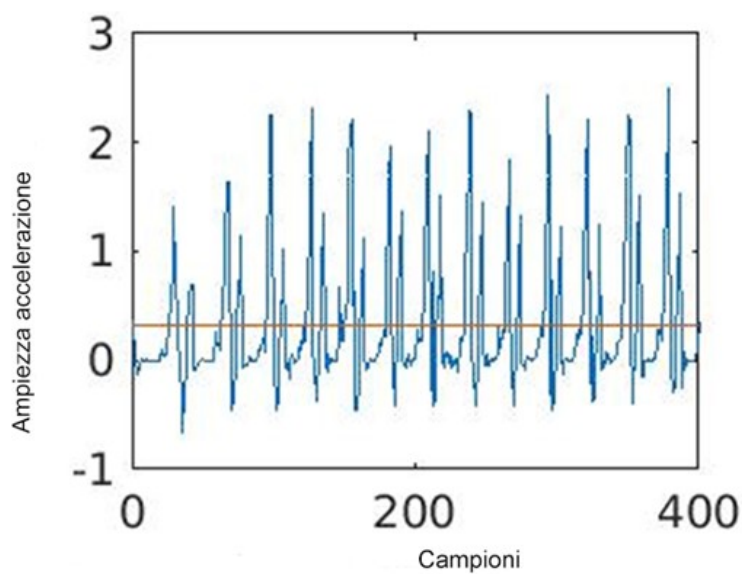


Fig. 19 Grafico accelerometrico tapis roulant con retta mean

Lo stesso lavoro viene effettuato per le altre acquisizioni, come mostrato nelle figure 20-21-22.

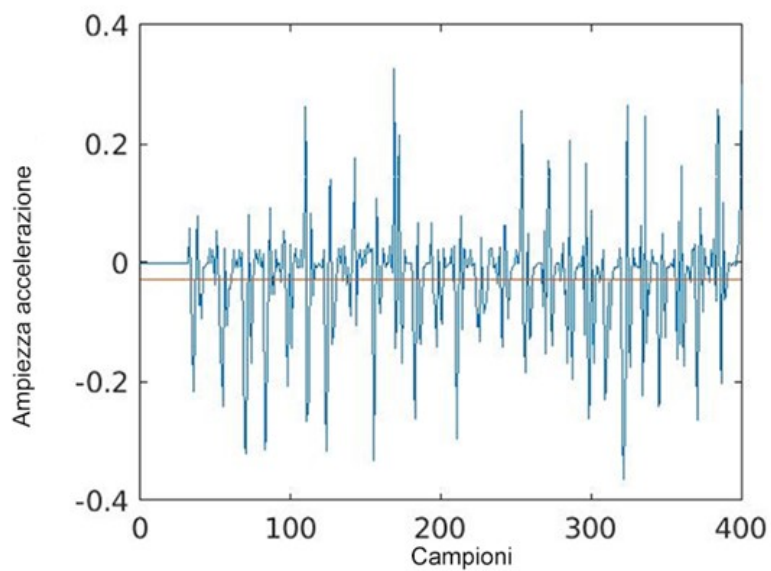


Fig. 20 Grafico accelerometrico piano rettilineo con retta mean

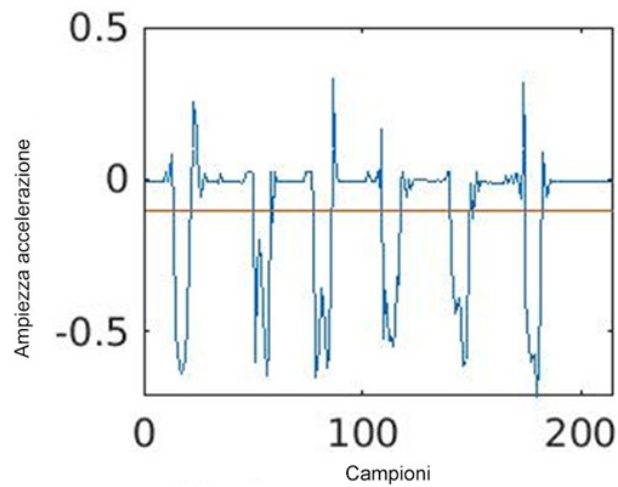


Fig. 21 Grafico salita con retta mean

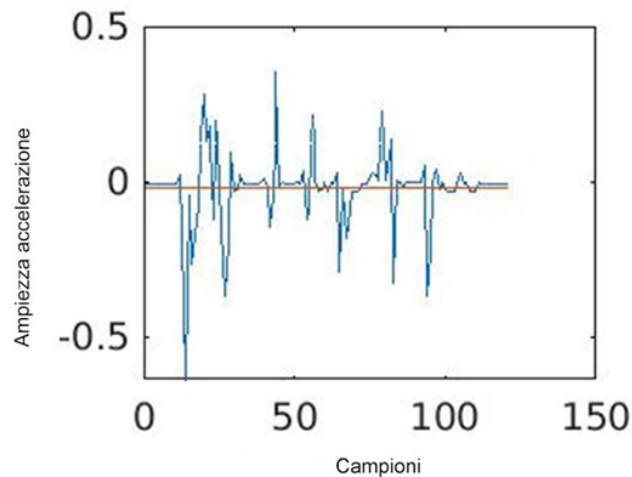


Fig.22 Grafico discesa con la retta mean

Dato che, come è ben visibile in tutti i grafici, la retta mean non interseca tutti i picchi, e quindi nel calcolo delle intersezioni con il segnale vengono esclusi i picchi minori, viene effettuata la sua traslazione, in modo da intercettare e includere nel conteggio un numero maggiore di picchi.

In particolare, si scelgono valori di traslazione della retta di **+0,75**, **+1** e **-1**, in modo da verificare quale sia la traslazione più idonea per l'analisi.

Il codice utilizzato per la traslazione della retta di **+0,75** è il seguente:

```
>> a_mean (:) = a_mean (:) + 0,75;
```

Fig. 23 mostra il grafico con le varie prove effettuate.

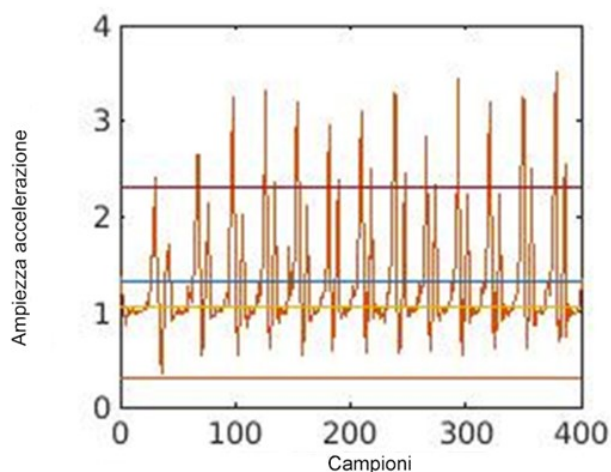


Fig. 23 Grafico tapis roulant con retta mean originaria (retta blu), traslata di + 0,75 (retta gialla), + 1 (retta rossa) e - 1 (retta arancione)

La scelta ricade sulla retta traslata di + 0,75, in quanto interseca anche i picchi minori del segnale.

A questo punto, si passa alla conta delle intersezioni della retta con il grafico.

```
>> g = a_mean > A_tapisroulant;
>> crossing = sum (xor (g (1:end-1), g (2:end)))
```

I risultati ottenuti sono stati confrontati con i rilevamenti fatti usando la app Steps mentre si indossava la calzatura, e vengono riportati nella seguente tabella.

Tabella 1. Dati raccolti dalle tre prove effettuate, confrontate con il numero dei passi rilevati con la app Steps.

	Rilevati	+0,75	+1	-1
Tapis Roulant	556 passi	1506/2 = 753 passi	864/2 = 432 passi	6/2 = 3 passi
Piano	583 passi	1430/2=715 passi	0	0
Salita	10 scalini	16/2 =8 scalini	0	0
Discesa	10 scalini	28/2 = 14 scalini	0	0

Analizzando le differenze tra i passi rilevati con la app e quelli delle tre soglie scelte per la retta mean, ipotizzando che un passo sia contato ogni due intersezioni della retta con il segnale, si è arrivati a scegliere come soglia migliore quella in +0,75, concludendo che anche i picchi minori rappresentano un passo.

Come è ben evidente però, l'algoritmo sbaglia in eccesso per il caso del Tapis Roulant, del Piano e della Discesa, mentre per la Salita abbiamo un numero di scalini in difetto.

3.3 Metodo rilevamento picchi con Findpeaks

Si è provato a migliorare la stima utilizzando il numero dei picchi di ogni segnale, ipotizzando che ogni picco corrisponda ad un passo. In questo modo, si cercherà di eliminare quelli che non rappresentano un vero passo, e quindi avere un numero ancora più vicino a quello rilevato all'inizio tramite la app Steps.

Si è impostato un valore minimo di ampiezza e di altezza di ciascun picco, in particolare di **0.5** per l'ampiezza, mentre per l'altezza si è scelto la **metà del massimo del valore del modulo dell'accelerazione** calcolato in precedenza.

Si riporta qui di seguito il codice utilizzato per l'analisi del segnale sul piano rettilineo.

```
>> minpeakwidth = 0.5;
>> minpeakheight = max (A_piano)/2;
>> [pks, locs] = findpeaks (Accelerazione, 'MinPeakWidth', minpeakwidth,
'MinPeakHeight', minpeakheight);
>> numpassi = length (pks);
>> fprintf ('Numero passi stimato con picchi %d\n', numpassi);
```

Il numero dei passi che il codice ci restituisce è pari circa alla metà dei passi, quindi ipotizziamo che l'altezza assegnata al picco non sia sufficiente, in quanto non considera tutti i picchi come passi. Per questo, essendo **2.4** il valore impostato con questa modalità, dimezziamo il valore:

```
>> minpeakheight = 0.12;
```

Il numpassi restituito in questo caso è di **593**, prossimo a quello rilevato con la app(**583**).

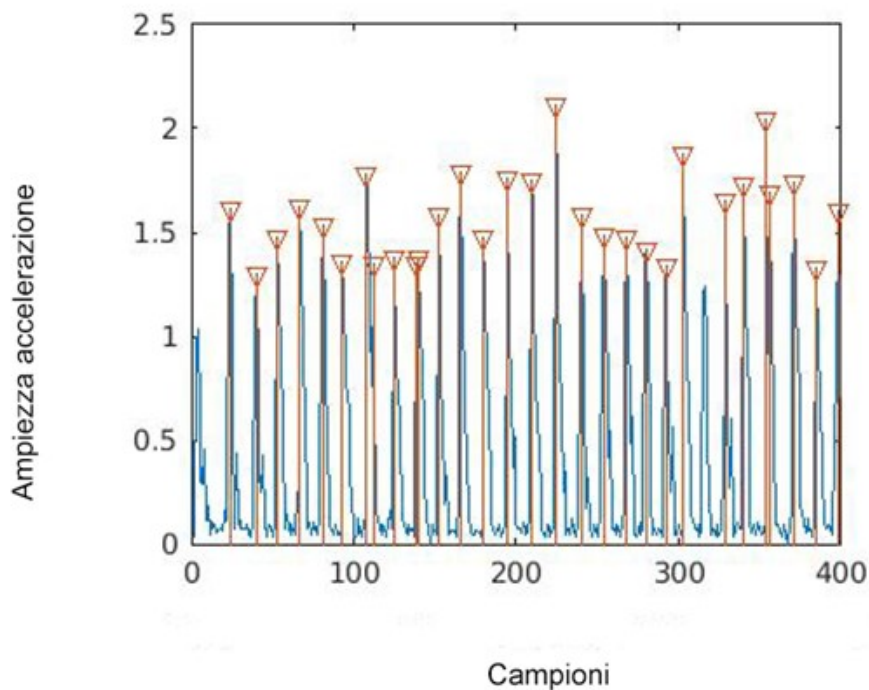


Fig.24 Grafico segnale piano di 400 campioni con rilevazione picchi

Come è possibile vedere in Figura 24, attraverso l'analisi dei picchi si è riusciti ad arrivare ad eliminare tutti quelli che risultano essere al di sotto dello 0.12, e che quindi possono essere eliminati dal conteggio dei passi.

Lo stesso lavoro è stato fatto per tutti gli altri segnali. Analizzando il segnale del Tapis Roulant, si è ottenuto il numero dei passi impostando anche in questo caso valori che potessero escludere i picchi non costituenti un passo.

Partendo anche in questo caso dal valore dell'altezza come la metà del massimo del segnale del modulo accelerazione, si è ottenuto un numero di picchi pari a **24**, valore troppo basso rispetto ai **556** passi rilevati con la app, e quindi si ipotizza che i picchi considerati non siano abbastanza.

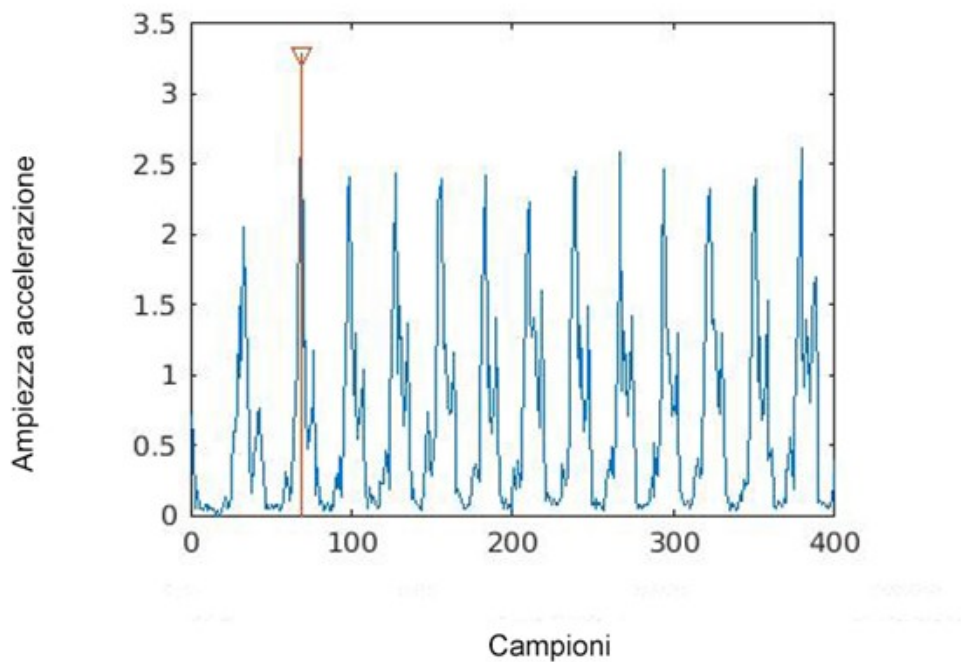


Fig. 25 Grafico tapis roulant con $\text{minpeakheight} = \max(A_{\text{tapis}})/2$

Si imposta un valore minore per minpeakheight , in modo da comprendere un numero di picchi maggiore.

>> $\text{minpeakwidth} = 0.5;$

>> $\text{minpeakheight} = 1.45;$

>> $\text{numpassi} = \text{length}(\text{pks});$

Il numero restituito è **260**, che è prossimo alla metà di quello registrato con la app (**556**).

Viene abbassato ancora il valore:

>> $\text{minpeakheight} = 0.38;$

Il numero passi restituito in questo caso è di **556**, che è esattamente il valore registrato.

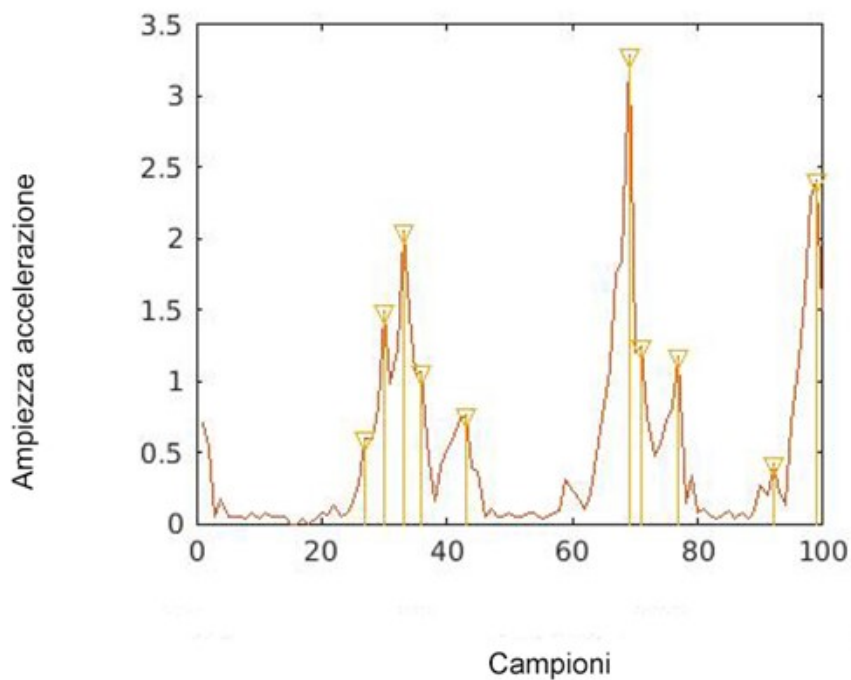


Fig.26 Grafico tapis roulant di 100 campioni con rilevamento picchi

Si passa poi al segnale per la conta degli scalini in discesa. In questo caso, impostando:

```
>> minpeakheight = max (Accelerazione)/2;
```

```
>> minpeakwidth = 0.5;
```

```
>> numscalini = length (pks);
```

si ottiene un numero di scalini pari a **11**, valore accettabile dato che gli scalini utilizzati per il segnale sono **10**.

Provando a modificare il *minpeaksheight*, alzando e abbassando il valore, il numero dei picchi considerati risulta insufficiente o troppo elevato, quindi si ipotizza che il valore scelto per il *minpeaksheight* sia adeguato.

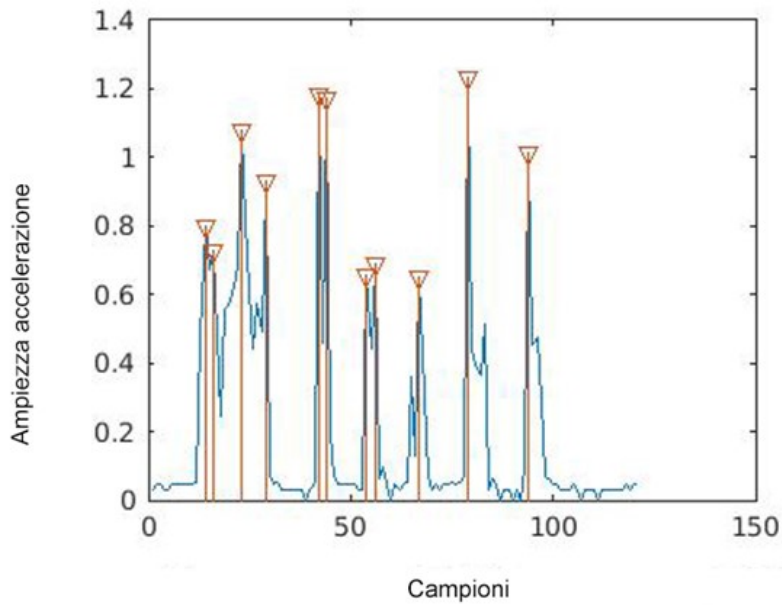


Fig. 27 Grafico discesa scalini con rilevazione picchi

Infine si analizza il segnale della salita sui **10** scalini.

Utilizzando lo stesso codice della salita, ovvero impostando l'altezza del picco come la metà del massimo del modulo dell'accelerazione, il numero di scalini restituito è pari a **12**.

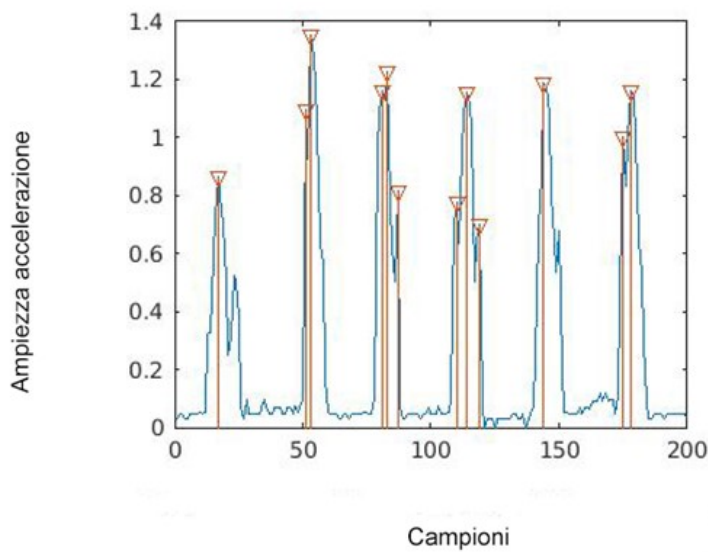


Fig.28 Grafico segnale salita con $minpeaksheight = \max(A_{salita})/2$

Il valore risulta prossimo al numero di scalini, ma si abbassa comunque il valore dell'altezza del picco in modo da provare a eliminare i picchi in eccesso.

Si sceglie:

>> *minpeakheight* = 0.79;

Il valore restituito del numscalini è proprio **10**, lo stesso degli scalini contati durante l'acquisizione del segnale.

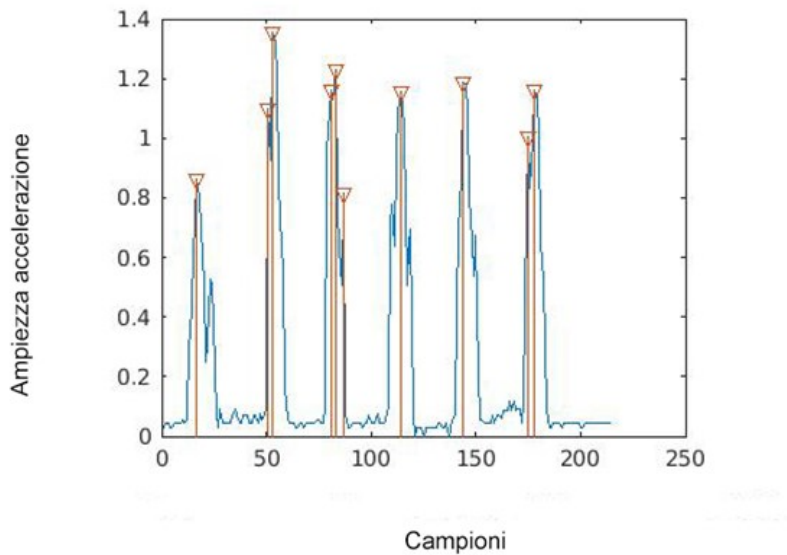


Fig.29 Grafico segnale *salita* con *minpeaksheight*= 0.79

In Tabella 2 riassumiamo il numero dei picchi rilevato.

Tabella 2. Numero picchi registrato sulle quattro acquisizioni.

	PICCHI
Tapis Roulant	556
Piano	593
Salita	10
Discesa	11

3.4 Confronto tra i due metodi

Confrontando i due metodi utilizzati e i dati ottenuti, si evince che i picchi intersecati dalla retta mean risultano essere in numero eccessivo (nel caso di Piano, Tapis Roulant e Discesa), in quanto la retta interseca picchi che non devono essere considerati passi, o insufficienti (nel caso della Salita) in quanto le intersezioni non comprendono anche i picchi che invece devono essere considerati.

Grazie alla rilevazione del numero di picchi, è stato però possibile stabilire un minimo valore che rappresenta l'altezza del picco, in modo da inglobare anche quelli che, con l'algoritmo precedente, non erano stati considerati, e a escludere quelli in eccesso.

Per evidenziare le differenze tra i due metodi utilizzati, si sono raccolti i dati ottenuti nella Tabella 3 qui di seguito riportata.

Tabella 3. Confronto tra i dati ottenuti dai due metodi utilizzati e quelli rilevati con la app Steps.

	Rilevati	Retta mean + 0,75	Numero picchi con findpeaks
Tapis Roulant	556 passi	753 passi	556 passi
Piano	583 passi	715 passi	593 passi
Salita	10 scalini	8 scalini	10 scalini
Discesa	10 scalini	14 scalini	11 scalini

CAPITOLO 4

Elaborazione dei segnali di pressione in Matlab

4.1 Acquisizione delle componenti del segnale di pressione

Un altro metodo utilizzato per la stesura di un algoritmo per il conteggio dei passi si basa sull'utilizzo dei segnali di pressione generati da sensori FSR, esposto in ^[14].

Come fatto in precedenza per i segnali accelerometrici, si sono elaborati mediante Matlab le tre componenti della pressione, S0, S1 e S2, di cui si riporta un esempio in Fig. 30, 31 e 32.

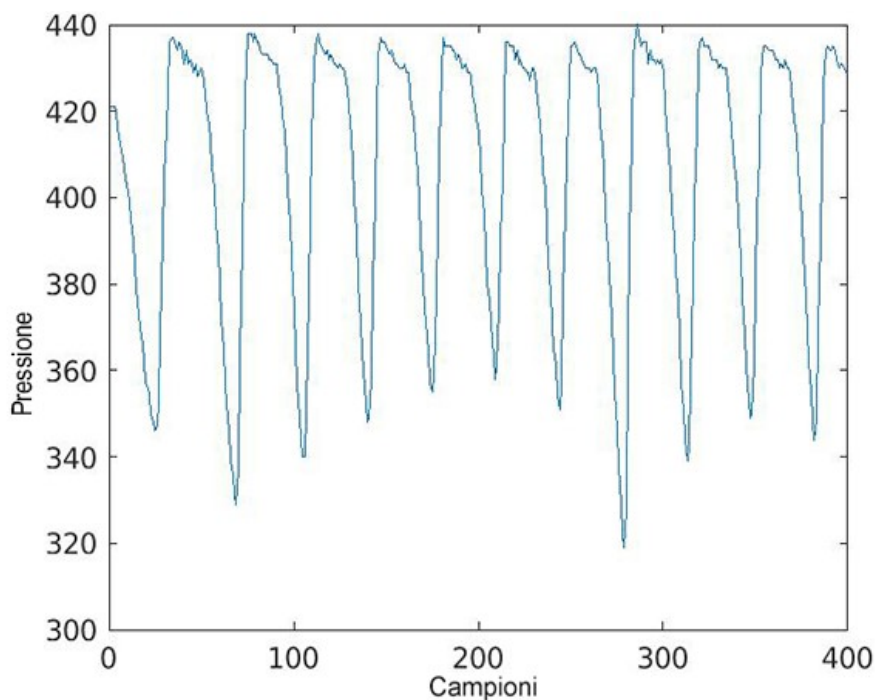


Fig. 30 Segnale S0 Tapis Roulant

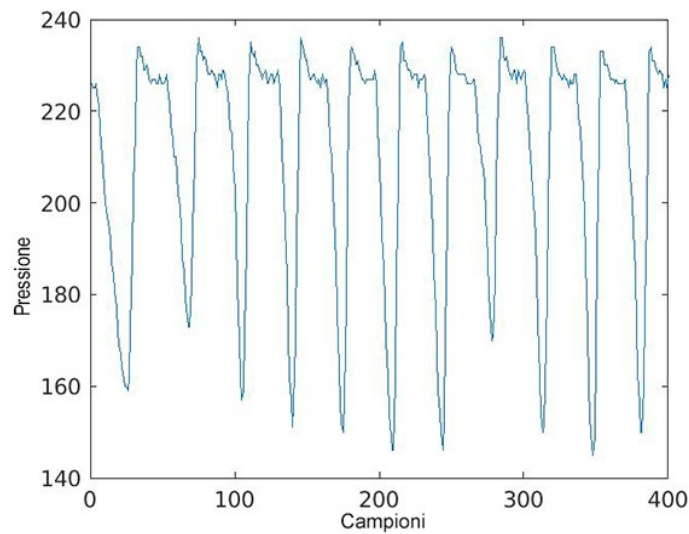


Fig.31 Segnale S1 Tapis Roulant

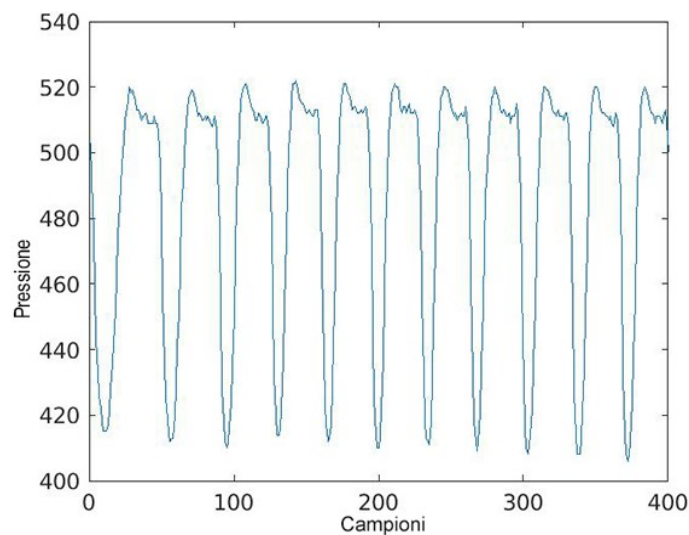


Fig.32 Segnale S2 Tapis Roulant

4.2 Composizione del segnale di Pressione Media

Successivamente, si è elaborato il segnale della pressione come media delle tre componenti:

$$\Delta P_{avg} = (S0 + S1 + S2)/3.$$

Si riportano di seguito i grafici in Fig. 33, 34, 35 e 36. In asse verticale è riportato il valore medio in uscita dall'ADC (Analog to Digital Converter).

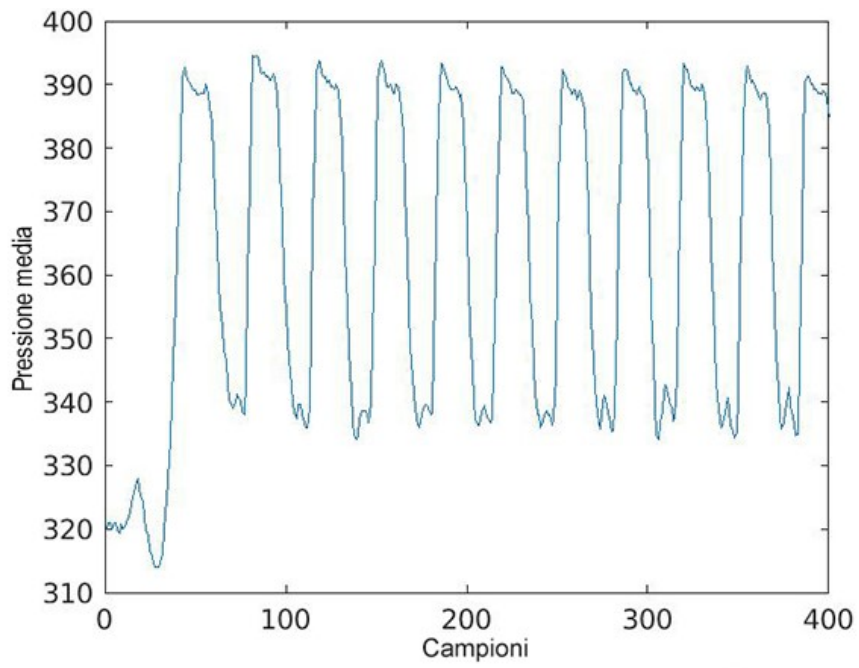


Fig.33 Segnale Pressione Media camminata su piano

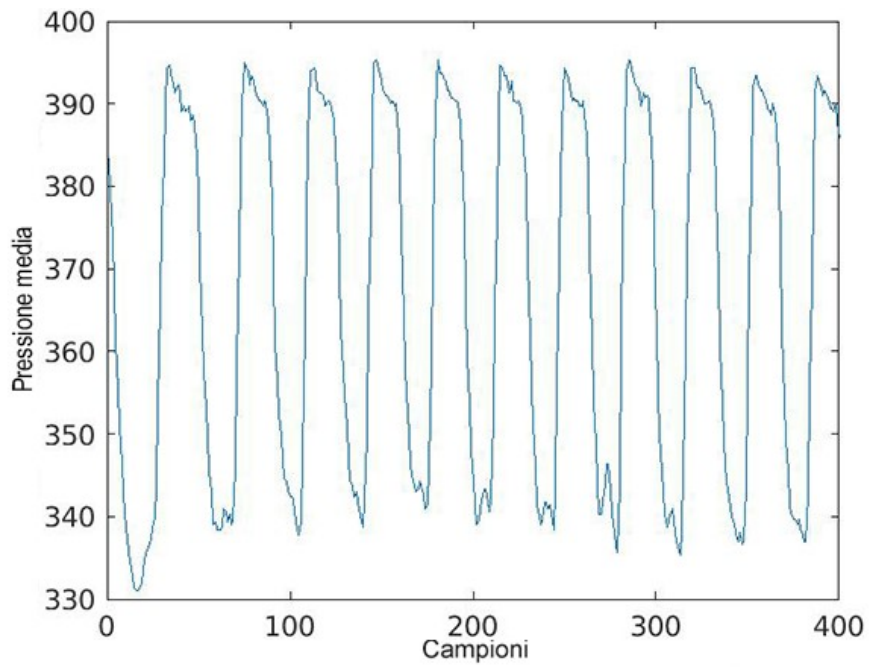


Fig.34 Segnale Pressione Media Tapis Roulant

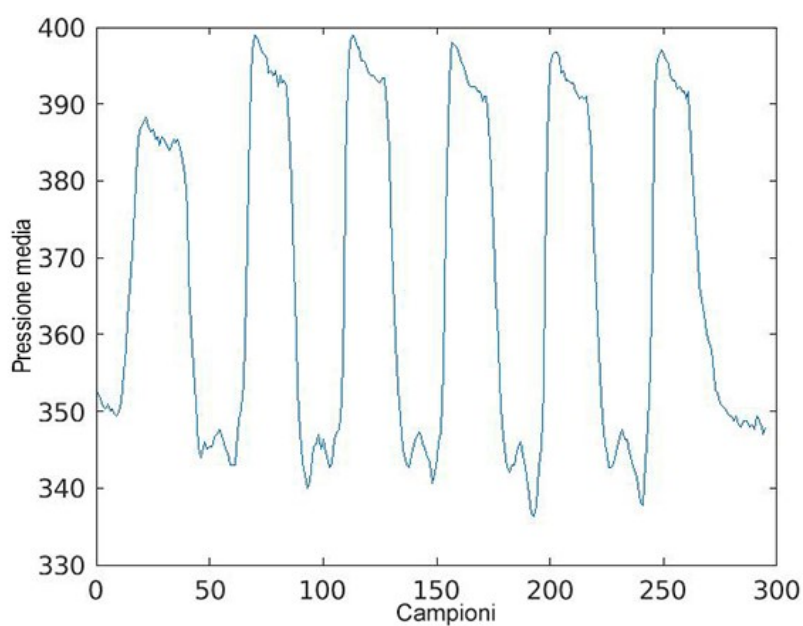


Fig.35 Segnale Pressione Media salita scale.

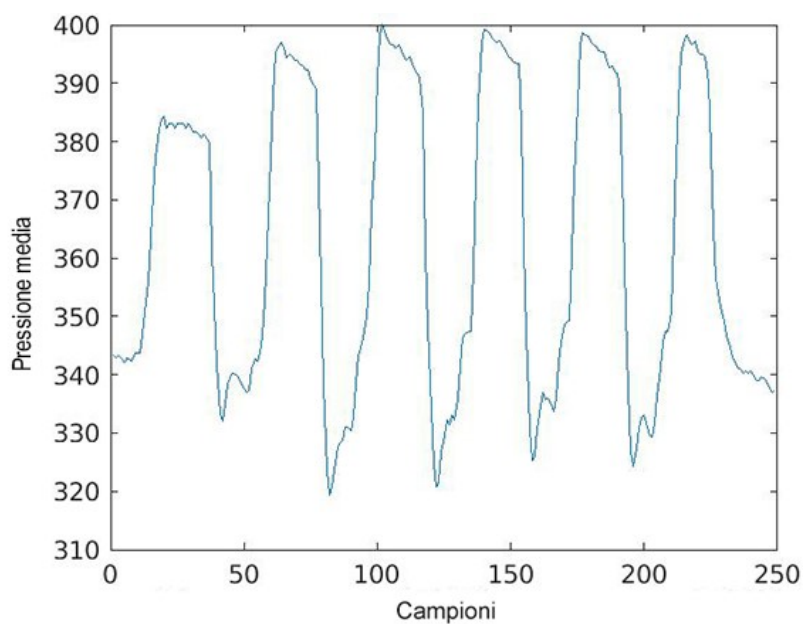


Fig. 36 Segnale Pressione Media discesa scale.

Come si può ben notare dai grafici appena mostrati, i segnali di pressione di tutte le acquisizioni mostrano un andamento quasi costante, quindi è possibile utilizzare il metodo sfruttato in precedenza per i segnali accelerometrici, ovvero il rilevamento dei picchi per il conteggio dei passi.

Conclusioni

Lo studio di un metodo per il conteggio dei passi è un argomento molto attuale e nessuno di quelli proposti fino ad ora può essere considerato universale.

Il motivo di questa particolarità di tutti i contapassi risiede nelle loro specifiche caratteristiche e nel loro ambito di utilizzo.

In generale però, si è potuto constatare che il metodo più semplice e immediato è incentrato sul rilevamento dei picchi, i quali, si stima, possano corrispondere a un passo.

Il lavoro svolto si è basato proprio su questo metodo, andando a verificare quale fosse il modo più accurato per la rilevazione dei picchi e, quindi, la stima dei passi effettuati in varie modalità di andatura.

Maggiori difficoltà sono state riscontrate nel conteggio dei passi tramite l'intersezione della retta mean con i segnali accelerometrici, in quanto la retta non ha intersecato tutti i picchi e il conteggio è risultato essere impreciso. Per ovviare al problema si è traslata la retta più volte, arrivando ad identificare una posizione che riuscisse a intersecare il maggior numero di picchi, modalità che comunque non ha fornito risultati accurati.

Il metodo con *findpeaks*, invece, ha dimostrato maggiore precisione, riuscendo a trovare una corrispondenza tra i picchi rilevati e i passi contati, anche se, per ogni modalità di andatura, si è dovuto adattare l'algoritmo rendendo quindi più complesso il lavoro.

Una valida alternativa più semplice e immediata può risultare il rilevamento dei segnali di pressione tramite i sensori FSR, in quanto i segnali risultano essere, all'incirca, costanti. Ovviamente, questo tipo di metodo può essere utilizzato solo per contapassi che comprendono sensori di forza, come nel caso della smartshoe utilizzata per questo lavoro.

Per altre tipologie di dispositivi indossabili, come smartwatch o smartphone che non sono dotati di sensori di forza, potrebbe però risultare interessante considerare i giroscopi per l'analisi dei segnali, analizzando, oltre l'accelerazione, anche la velocità angolare. Questo tipo di dispositivi infatti, a differenza delle smartshoes, durante la deambulazione seguono il movimento delle braccia di chi cammina (vengono tenuti in mano nel caso dello smartphone, o sul polso nel caso dello smartwatch), andando quindi a registrare un movimento rettilineo e un movimento rotazionale. Ipotizzando che la

sbracciata sia collegata nella maggior parte dei casi al movimento delle gambe in avanti, si potrebbe tener conto di entrambi i segnali dell'accelerazione e della velocità angolare per il conteggio dei passi.

Bibliografia

- [1] Wei International Conference on Computer Systems, Electronics and Control (ICCSEC), 2017 Yanxin, Fan Xiujuan ,Yang Dongchen. *Algorithm Design of Wearable Pedometer.*
- [2] IEEE Explore Article, A. Wang, X. Ou, B. Wang, *Improved Step Detection And Step Length Estimation Based On Pedestrian Dead Reckoning*
- [3] Sensors Article, 2013 Melania Susi, Valérie Renaudin, Gérard Lachapelle. *Motion Mode Recognition and Step Detection Algorithms for Mobile Phone Users*
- [4] Sensors Article, 2016 Mingrong Ren, Kai Pan, Yanhong Liu, Hongyu Guo, Xiaodong Zhang, PuWang. *A Novel Pedestrian Navigation Algorithm for a Foot-Mounted Inertial-Sensor-Based System.*
- [5] Mobicom '12: Proceedings of the 18th annual international conference on Mobile computing and networking, 2012 A. Rai, K. K. Chintalapudi, V. N. Padmanabhan and R. Sen, *Zee:zero-effort crowdsourcing for indoor localization.*
- [6] UbiComp'13, Zurich, Switzerland. 2013 Agata Brajdic, Robert Harle. *Walk Detection and Step Counting on Unconstrained Smartphones.*
- [7] Sensors Article, 2018 College of Computer Science, Inner Mongolia University, Hohhot 010021, China. Xiaomin Kang, Baoqi Huang, Guodong Qi, *A Novel Walking Detection and Step Counting Algorithm Using Unconstrained Smartphones.*
- [8] Article, Sigma J Eng & Nat Sci 2017, A. C. DİRİCAN, S. AKSOY, *Step Counting Using Smartphone Accelerometer And Fast Fourier Transform.*
- [9] IEEE SENSORS JOURNAL, VOL. 16, NO. 21, NOVEMBER 1 2016, Yunhoon Cho, Hyuntae Cho, and Chong-Min Kyung, *Design and Implementation of Practical Step Detection Algorithm for Wrist-Worn Devices.*

[10] Conference published in IEEE Explorer 2018, Xiaomin Kang ; Baoqi Huang ; Runze Yang ; Guodong Qi, *Accurately Counting Steps of the Pedestrian with Varying Walking Speeds.*

[11] UBICOMP '16, SEPTEMBER 12–16, 2016, HEIDELBERG, GERMANY Baoqi Huang, Guodong Qi, Xiaokun Yang, Long Zhao, Han Zou, *Exploiting Cyclic Features of Walking for Pedestrian Dead Reckoning with Unconstrained Smartphones.*

[12] Article Computer Communications, journal homepage: www.elsevier.com/locate/comcom, Hui Sun, Qiong Wu, *Detection algorithm of regional peak motion based on acceleration sensor.*

[13] Publication: UbiComp '09: Proceedings of the 11th international conference on Ubiquitous computing, 2009 J. Lester, C. Hartung, L. Pina, R. Libby, G. Borriello and G. Duncan, *Validated caloric expenditure estimation using a single body-worn sensor.*

[14] IEEE Explorer, 2018 Satetha Siyang, Thirawut Nilpanapan, Teerakiat Kerdcharoen, *Development of IoT-Based Data Shoes for Daily Life Step Count.*

[15] IEEE Explorer, 2019 Yanhu Huang , Lvqing Bi , Wentao Fu , Ruizhao Yang , Xueguang Bi, **A precise step counting algorithm based on acceleration correlation analysis.**

Ringraziamenti

Al termine di questo importante percorso, vorrei ringraziare tutti coloro che mi hanno sostenuto in questi anni, dandomi il loro supporto sia nei momenti piacevoli, che in quelli più difficili.

In primis, vorrei ringraziare la Professoressa Susanna Spinsante che, con pazienza e meticolosità, mi ha guidato in questo lavoro, dispensando consigli e aiuti essenziali, e la Professoressa Stefania Cecchi, che mi ha sempre spronato e incoraggiato.

Il più importante ringraziamento va alla mia famiglia, senza la quale non avrei mai potuto raggiungere questo traguardo. In particolare, ringrazio mamma Tina per la pazienza e per non avermi mai negato nulla, dandomi anche più di quel che meritavo. Ringrazio papà Lanfranco per avermi sempre aiutato, di giorno e di notte, quando ne avevo bisogno, permettendomi di arrivare a essere, dopo di lui, il secondo Ingegnere della famiglia.

Un ringraziamento va anche a mia sorella Dalila, compagna e amica di vita, e alla più piccola della famiglia Debby, che mi ha aiutato nella parte grafica di questo lavoro, dandomi utili consigli.

Indispensabili e insostituibili sono la forza e il sostegno datomi dal mio compagno di vita Ivan, che, con amore incondizionato, mi ha accompagnato negli ultimi anni di questo percorso, con l'auspicio che questo possa essere solo l'inizio di un lungo cammino insieme.

Un caloroso ringraziamento va anche alle nonne Angela e Ada che, attraverso i loro racconti di un'epoca molto lontana dalla mia, mi hanno fatto comprendere l'importanza dello studio e dei valori per un futuro migliore. Non dimentico anche lo zio Mario, che mi ha sempre seguito in tutti i miei esami.

Un ringraziamento speciale va ai miei più cari amici: Giorgia, la mia "coinquilina preferita", ma soprattutto amica/sorella con cui ho condiviso gioie e dolori, e Fabio, amico sempre pronto ad aiutarmi in qualsiasi momento del giorno e della notte.

Infine, ringrazio i miei secondi genitori, Lorenza e Rodolfo, che mi hanno accolta fin da subito come una figlia, la zia Catia e le mie “nonne adottive” Maria e Gina, che ormai considero come la mia seconda famiglia.

Ringrazio chiunque sia arrivato a leggere questo mio lavoro fin qui e, nella speranza che questo sia solo l'inizio di un lungo percorso fatto di soddisfazioni e gioie, vi abbraccio tutti con affetto.