



**UNIVERSITÀ POLITECNICA DELLE MARCHE**  
**FACOLTÀ DI INGEGNERIA**

---

Corso di Laurea Triennale in Ingegneria Biomedica

**Acquisizione e analisi di segnali EMG e inerziali acquisiti mediante  
sensori indossabili durante diverse modalità di cammino**

**Acquisition and analysis of EMG and inertial signals acquired using wearable sensors  
during different walking modes**

**RELATORE**

Dr. Andrea Tigrini

**TESI DI LAUREA DI:**

**CORRELATORI**

Prof.ssa Federica Verdini

Marta Palazzoli

Prof. Alessandro Mengarelli

**Anno Accademico 2023/2024**



# INDICE

<b>Introduzione</b>	<b>7</b>
<b>Obiettivo dell'elaborato</b>	<b>10</b>
<b>Stato dell'arte</b>	<b>13</b>
1.1 Stato dell'arte	13
1.2 Il ciclo del passo	16
1.3 Individuazione delle fasi del ciclo del passo	19
1.4 Myo Armband	20
1.5 Unità di misura inerziale	21
1.6 Elettromiografia	25
<b>Metodi</b>	<b>27</b>
2.1 Partecipanti	27
2.2 Protocollo di misura	29
2.3 Elaborazione dei dati	33
2.4 Rule-based methods	35
2.5 Camminata in piano	36
2.6 Camminata su rampa	40
2.7 Salita e discesa di scale	41
2.8 Elaborazione del segnale elettromiografico	43
2.9 Organizzazione del dataset	45

2.10	Implementazione degli algoritmi	47
	<b>Risultati</b>	<b>52</b>
3.1	Creazione del data set	52
3.2	Segmentazione delle fasi del ciclo del passo	54
	<b>Discussioni</b>	<b>62</b>
	<b>Conclusioni</b>	<b>66</b>
	<b>Appendice - Codici Matlab</b>	<b>69</b>

## **ELENCO DELLE FIGURE E DELLE TABELLE**

Figura 1. Myo Armband.	14
Figura 2. Fasi del ciclo del passo.	18
Figura 3. Assi dell'IMU nel Myo Armband.	24
Figura 4. Visualizzazione grafica del dispositivo indossato sulla gamba.	29
Figura 5. Sub01, velocità angolare acquisita nelle tre prove (camminata in piano, su rampa, salita e discesa di scale).	31
Figura 6. Sub01, dettaglio dei primi cicli del passo della velocità angolare acquisita nelle tre prove (camminata in piano, su rampa, salita e discesa di scale).	31
Figura 7. Sub01, EMG acquisito durante la camminata in piano.	32
Figura 8. Sub01, EMG acquisito durante la camminata su superficie inclinata.	32
Figura 9. Sub01, EMG acquisito durante salita e discesa di scale.	33
Figura 10. Individuazione dei punti di zero-crossing.	37
Figura 11. Individuazione dei punti IC.	38
Figura 12. Velocità angolare medio laterale di un tipico ciclo del passo, in cui si evidenziano gli eventi IC, HR, TO, FA, TBV.	39

Figura 13. Componente medio laterale della velocità angolare durante il cammino su superficie inclinata e individuazione delle fasi di MS, HS, TO.	41
Figura 14. Segnale relativo al giroscopio durante salita (figura a sinistra) e discesa (figura a destra) di scale con i relativi punti IC e FO.	42
Figura 15. Organizzazione dei dati acquisiti in formato Excel.	48
Figura 16. Modello di tabella in formato Excel relativa alla segmentazione delle prove eseguite in piano.	49
Figura 17. Modello di tabella in formato Excel relativa alla segmentazione delle prove eseguite su piano inclinato.	50
Figura 18. Modello di tabella in formato Excel relativa alla segmentazione delle prove eseguite su scale.	51
Figura 19. Rappresentazione grafica del dataset completo.	53
Figura 20. Esempio di individuazione delle fasi del ciclo del passo per le prove eseguite in piano.	54
Figura 21. Dettaglio di individuazione delle fasi nei primi cicli del passo per le prove eseguite in piano.	55
Figura 22. Esempio di individuazione delle label sovrapposte al segnale per le prove eseguite in piano.	56

Figura 23. Esempio di individuazione delle fasi del ciclo del passo per le prove eseguite su piano inclinato.	57
Figura 24. Dettaglio di individuazione delle fasi nei primi cicli del passo per le prove eseguite su piano inclinato.	57
Figura 25. Esempio di individuazione delle label sovrapposte al segnale per le prove eseguite su piano inclinato.	58
Figura 26. Esempio di individuazione delle fasi del ciclo del passo per le prove eseguite su scale.	59
Figura 27. Dettaglio di individuazione delle fasi nei primi cicli del passo per le prove di salita di scale.	59
Figura 28. Dettaglio di individuazione delle fasi nei primi cicli del passo per le prove di discesa di scale.	60
Figura 29. Esempio di individuazione delle label sovrapposte al segnale per le prove eseguite su scale.	60
Figura 30. Accuratezza della classificazione delle fasi del ciclo del passo utilizzando il segnale EMG.	65
Tabella 1. Caratteristiche dei partecipanti.	28



# Introduzione

Nel 2017 è stato stimato che 57,7 milioni di persone nel mondo convivono con l'amputazione degli arti inferiori dovuta a cause traumatiche. Le malattie vascolari periferiche e il diabete sono comunemente identificate come le principali cause di amputazione di paesi più sviluppati. Tuttavia, nei paesi in via di sviluppo, le cause principali sono principalmente traumi (incidenti stradali, cadute e infortuni) [1].

In Italia, attualmente, riportando quanto detto da Alberto Momoli (direttore dell'Unità operativa di Ortopedia e Traumatologia dell'ospedale di Vicenza e presidente della SIOT, Società Italiana di Ortopedia e Traumatologia), si stimano che siano all'incirca 15 mila le amputazioni dell'arto inferiore che vengono eseguite ogni anno [2]. Perciò è importante, ove possibile, utilizzare componenti protesiche che possono imitare, con diversi livelli di complessità, il movimento dell'arto che vanno a sostituire.

Nel corso dell'ultimo decennio, è stata sviluppata un'intera generazione di protesi transtibiali. Tuttavia, queste tecnologie sono limitate dal rilevamento delle fasi del ciclo del passo che controllano il dispositivo in funzione dell'andatura di chi lo indossa al fine di riprodurre la corrispondente funzionalità durante il cammino. Di conseguenza, è di fondamentale importanza andare ad utilizzare le giuste tecniche al fine di individuare, con quanta più precisione possibile, le diverse fasi che contraddistinguono il cammino dell'uomo [3].

Nonostante i numerosi progressi tecnologici degli ultimi anni, molte persone che hanno subito un'amputazione degli arti inferiori sono limitate all'utilizzo di protesi passive che, però, comportano una camminata asimmetrica del paziente che le indossa e un maggior dispendio energetico. Questo limita molto l'utente, creando difficoltà in attività come la salita di rampe o di scale, che richiedono un maggior sforzo energetico, spesso non garantito dalle protesi passive [4]. Per affrontare questa problematica, gli studi più innovativi prendono in considerazione lo sviluppo e le tecniche che possono risultare utili nella costruzione delle protesi attive che, rispetto a quelle passive che non sono controllate da sensori, permettono andature più naturali e una gamma più ampia di movimenti, fornendo una propulsione attiva per il paziente che le indossa [5]. Le protesi attive integrano sensori che forniscono feedback ad un controller, il quale regola un attuatore. La funzionalità del controller si basa sulla determinazione dello stato attuale dell'andatura e sulla formulazione di una strategia di risposta adeguata in tempo reale [6]. Esistono diversi tipi di sensori che sono integrati nella protesi per rilevare correttamente le varie fasi che contraddistinguono l'andatura ed attuare delle risposte che garantiscono la corretta deambulazione; tra questi i più usati sono i sensori elettromiografici (EMG) o le unità di misura inerziali (IMU). I sensori EMG hanno la potenzialità di raggiungere un elevato range di controllo volontario sulla protesi, mentre l'utilizzo di IMU permette l'identificazione delle varie fasi che compongono il ciclo del passo [4]. Lo sviluppo di protesi attive rimane però una sfida impegnativa per quanto riguarda il controllo elettromiografico poiché il segnale EMG risulta essere poco stabile soprattutto a lungo termine e può causare quindi un rischio per il paziente che indossa una protesi controllata esclusivamente da esso [7]. C'è quindi il bisogno di sviluppare nuovi algoritmi di controllo che possano essere efficaci e sicuri per l'utente, integrando anche i dati forniti dal sensore inerziale.

Oltre a ciò, è necessario condurre studi che forniscano una maggiore quantità di informazioni utili in questo campo, che considerino l'analisi del cammino in varie situazioni di quotidianità e che utilizzino un set up semplice. Fare affidamento su più sensori comporta la gestione di più parametri di controllo e l'aumento del rischio di guasti, oltre a limitare i movimenti di chi li indossa. Risulta, quindi, fondamentale condurre studi che esplorino soluzioni efficaci con un numero ridotto di sensori.

I recenti progressi hanno portato allo sviluppo di sensori indossabili che, oltre ad essere affidabili e convenienti, consentono il loro utilizzo anche al di fuori dell'ambiente di laboratorio e di poter svolgere l'analisi anche durante le attività della vita quotidiana. L'integrazione di questi nuovi dispositivi con avanzate tecniche di elaborazione e gestione dei dati che permettono lo sviluppo di nuovi dataset, rappresenta un importante contributo alla ricerca in questo ambito, che ha come obiettivo finale quello di migliorare le condizioni di vita di persone che utilizzano componenti protesiche.

## Obiettivo dell'elaborato

Il seguente elaborato si propone di presentare un protocollo sperimentale che ha come obiettivo la raccolta di dati in diverse condizioni di locomozione (camminata in piano, su rampa e salita/discesa di scale) tramite un dispositivo indossabile, il Myo Armband, dotato di otto elettrodi elettromiografici (EMG) ed un'unità di misura inerziale (IMU). I risultati sono raccolti ed organizzati in un dataset che può fungere da nuova risorsa per esplorare algoritmi e modelli innovativi, potenzialmente utili nella ricerca protesica. Risulta essere, infatti, di fondamentale importanza fornire dati relativi agli arti inferiori che siano il più completi possibili, al fine di promuovere la ricerca, lo sviluppo e l'innovazione nei settori correlati. L'organizzazione dei dati e delle informazioni raccolte dal sensore è stata effettuata facendo riferimento allo studio condotto da Wei et al. (2023) [8] che fornisce un dataset che rappresenta un modello di organizzazione di informazioni motorie per la decodifica dei movimenti che interessano gli arti inferiori.

Nel seguente elaborato, per ogni modalità di cammino per la quale sono stati svolti i test, vengono presentati degli algoritmi che permettono di segmentare il ciclo del passo e identificare le fasi relative ad esso. L'elaborazione e l'analisi dei segnali acquisiti si basano sulle tecniche e sui modelli innovativi che risultano fondamentali per l'identificazione dei vari eventi che compongono la camminata. Una rilevazione accurata delle fasi del passo è cruciale per il successo delle protesi per gli arti inferiori. Con l'incremento della versatilità e della complessità di questi dispositivi medici, infatti, cresce l'interesse nella ricerca per

rendere gli algoritmi di rilevazione del passo più efficienti e i sensori più compatti e facilmente indossabili. Un algoritmo di rilevazione funzionale, insieme ad un dataset ben organizzato, potrà incrementare significativamente la precisione, la stabilità e la sicurezza delle protesi e di altri dispositivi riabilitativi.

A differenza delle protesi utilizzate per gli arti superiori, non esistono, per ora, in commercio protesi per gli arti inferiori che integrano i segnali acquisiti dal sistema neuromuscolare dell'utente che le indossa [7]. Le protesi mioelettriche prendono in considerazione i segnali EMG relativi ai muscoli della porzione di arto rimanente per controllare i movimenti da attuare. L'elevato numero di amputazioni degli arti inferiori, unito alla carenza di protesi controllate dal sistema neuromuscolare, evidenzia l'importanza di studi finalizzati alla raccolta di un maggior volume di dati, includendo modalità di cammino quotidiane e combinando tecniche consolidate con nuovi algoritmi, che potrebbero rappresentare una svolta decisiva in questo campo di ricerca.



# Capitolo 1

## Stato dell'arte

### 1.1 Stato dell'arte

L'analisi del cammino è uno studio sistematico che riguarda la misura, la descrizione e l'individuazione di quantità che caratterizzano la locomozione dell'uomo. Attraverso lo studio del cammino umano, è possibile individuare le fasi che caratterizzano la deambulazione, determinare i parametri cinetici e cinematici e valutare quantitativamente le funzioni muscoloscheletriche. Le prime ricerche inerenti a questo ambito risalgono alla fine del 1800 e sono diventate di fondamentale importanza in campo diagnostico, terapeutico e riabilitativo [9]. È un settore ampiamente studiato e che ha importante applicazione in medicina in quanto permette l'osservazione e il monitoraggio di numerose patologie. Oltre all'ambito clinico e di laboratorio, lo studio del movimento può essere anche svolto in ambienti quotidiani, grazie allo sviluppo di nuovi dispositivi indossabili, che, oltre ad essere economici e facilmente utilizzabili, risultano anche essere efficienti e utilizzabili in ambienti più dinamici. Le tecnologie inerziali indossabili possono essere di grande utilità al fine di estrarre le caratteristiche inerenti al cammino dell'uomo, anche se comunque richiedono un utilizzo accorto da parte di chi le utilizza [10].

Attualmente esistono in commercio differenti dispositivi che possono essere facilmente utilizzabili anche dai meno esperti. Nello studio proposto nel seguente elaborato si utilizzerà

il Myo Armband che, nonostante sia stato progettato per essere indossato sul braccio, può essere anche posto sulla gamba, mantenendo le sue funzionalità parziali o complete. Ciò è stato dimostrato anche nel progetto svolto da A. Matos et al. (2016) [11], in cui i dati EMG e IMU sono stati acquisiti da tre individui che indossavano il bracciale su una gamba.



*Figura 1. Myo Armband.*

Myo è un dispositivo miografico full stack, dotato di sensori per acquisire dati inerziali e muscolari e dispone di un proprio kit di sviluppo software. Ciò potrebbe consentire lo sviluppo di soluzioni per le quali non era stato originariamente pensato. A tal proposito, il bracciale elettromiografico può essere utilizzato con successo anche per l'analisi degli arti inferiori, poiché la sua struttura circolare (come mostrato in **Figura 1** [12]) e la sensoristica presente sono adeguate al fine di misurare le attività coinvolte nell'analisi del ciclo del passo, andando ad acquisire, oltre al segnale elettromiografico, anche l'accelerazione e la velocità angolare del segmento corporeo coinvolto.

Attualmente non esistono molti dataset che riguardano il movimento degli arti inferiori, e la maggior parte sono limitati rispetto a diversi aspetti che ne impediscono l'ampio utilizzo. Ad esempio, molti non forniscono direttamente etichette dettagliate delle fasi del passo, che sono in realtà essenziali per condurre indagini più approfondite e precise [8].

Inoltre, tra i dataset disponibili, alcuni prevedono solo la camminata in piano [13] o su tapis roulant [14]. L'utilizzo di quest'ultimo introduce dinamiche che sono specifiche per il mantenimento dell'equilibrio durante il cammino su tapis roulant e che non si ripresentano fedelmente nel caso di cammino su superficie stabile e, quindi, non è consigliabile per effettuare studi volti ad effettuare un'analisi che si basa sulla quotidiana deambulazione dell'uomo [15], [16]. Altri studi, pur considerando diverse modalità di locomozione, impiegano un set up complesso che prevede l'utilizzo di più sensori posizionati in varie parti del corpo, il che può limitare la libertà di movimento e rendere difficile l'accesso o l'utilizzo in ogni ambiente [17], [18], [8].

L'analisi del cammino può essere condotta grazie alla sua natura ciclica, che permette l'identificazione di pattern che si ripetono in modo consecutivo. Sono stati, infatti, individuati diversi eventi che contraddistinguono il ciclo del passo. A tal fine, numerosi studi si sono concentrati sull'individuazione delle fasi di stance e swing [19], [20], che corrispondono rispettivamente ai momenti in cui l'arto è in appoggio e in oscillazione durante la camminata. Oppure, sono stati svolti studi volti ad individuare tre fasi [21], [22], [23], quattro [24], [10], cinque [25], [26], sei [27] e altri che coinvolgono ancora più fasi [28], [29].

Si è osservata inoltre che l'unità di misura inerziale (IMU) è il sensore maggiormente utilizzato nel rilevamento delle fasi e degli eventi. Questo è dovuto dal fatto che gli IMU sono particolarmente adatti per applicazioni a lungo termine e offrono numerosi vantaggi rispetto al segnale elettromiografico (EMG), tra cui un minor consumo energetico, una maggiore durata, un costo inferiore, un peso ridotto, portabilità e facilità di posizionamento sul corpo e un maggior numero di rilevamento delle fasi [30]. Sarebbe inoltre più opportuno

utilizzare il segnale offerto dal giroscopio in quanto si è visto che, per quanto riguarda il monitoraggio della camminata dell'uomo, fornisce prestazioni migliori rispetto alle altre quantità inerziali poiché presenta dei pattern definiti e confrontabili [20], [23]. Inoltre, il giroscopio non è influenzato dalla forza di gravità e funziona bene anche rispetto ad un posizionamento poco preciso [31], [20].

Considerando l'attuale stato dell'arte, c'è il bisogno di trovare soluzioni di organizzazione ed elaborazione dei dati versatili, efficaci, che siano confrontabili e che descrivano l'andatura dell'uomo sia in situazioni controllate che in circostanze di vita reale, utilizzando pochi sensori, nel tentativo di essere facilmente riproducibili in contesti diversi.

## **1.2 Il ciclo del passo**

La locomozione nell'uomo prevede un movimento semi-periodico, che coinvolge l'attuazione nel tempo di diversi muscoli che sono coordinati dal sistema nervoso centrale. In una persona che non ha particolari patologie, il ciclo del passo è definito come il periodo che va dal contatto iniziale del piede e il conseguente riappoggio dello stesso e, solitamente, viene diviso in due principali fasi: appoggio (*stance*) e sospensione (*swing*).

La fase di appoggio nella normale deambulazione occupa circa il 62% del ciclo del passo, ed è la fase in cui il piede è in contatto con il terreno. Mentre la fase di swing occupa circa il 38% e in questo periodo avviene la sospensione dell'arto che si prepara al successivo

appoggio [32]. All'aumentare della velocità, la fase di stance andrà ad accorciarsi, mentre quella di swing ad allungarsi durante il ciclo del passo.

Inoltre, la fase di appoggio può essere a sua volta divisa in [33]:

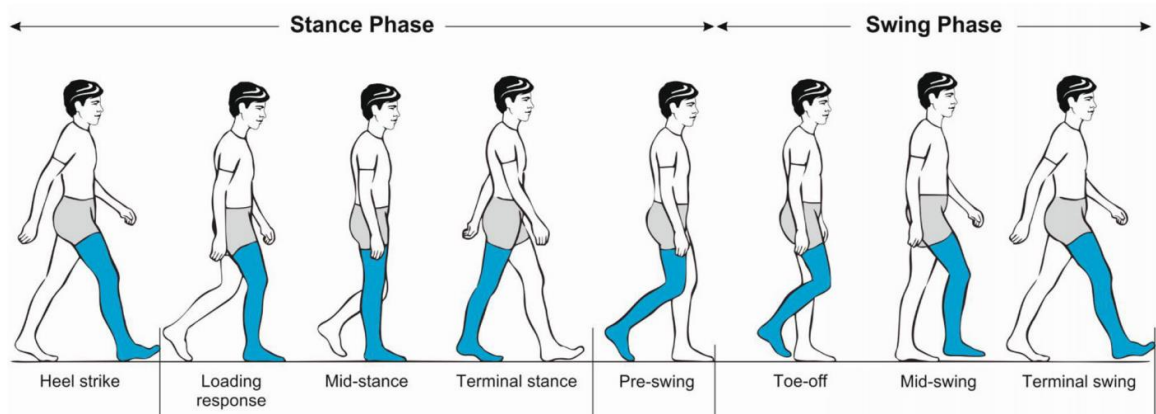
- *Initial contact (Heel contact)*: comprende il momento in cui il piede tocca il terreno; la postura con cui si compie questa fase, determinerà la modalità di risposta dell'arto all'appoggio. La gamba controlaterale sta terminando la fase di stance.
- *Loading response (Foot flat)*, 10% del ciclo del passo: il piede si appoggia interamente al terreno, inizia con il contatto iniziale e continua finché il piede controlaterale non si stacca dal suolo.
- *Mid-stance*, dal 10% al 30% del ciclo del passo: momento in cui l'arto controlaterale supera il piede d'appoggio. Questa fase continua finché il tallone ipsilaterale non si solleva da terra. Il corpo inizia a passare dall'assorbimento della forza ad una propulsione in avanti.
- *Terminal stance (Heel off)*, dal 40% al 50% del ciclo del passo: corrisponde al distacco del tallone, il peso del corpo si trova davanti all'avampiede, inizia quando il piede controlaterale tocca il suolo terminando la fase di swing e continua finché il piede ipsilaterale non si stacca dal suolo.
- *Pre-swing*, dal 50% al 60% del ciclo del passo: inizia con l'appoggio del piede controlaterale e termina quando il piede si stacca dal terreno.

La fase di oscillazione prevede le seguenti tre parti:

- *Toe Off (Acceleration)*, dal 60% al 75% del ciclo del passo: il piede è sollevato e la gamba viene portata avanti grazie alla flessione dell'anca e una crescente flessione del ginocchio.
- *Mid-swing*, dal 75% al 85% del ciclo del passo: prevede lo spostamento dell'arto da una posizione posteriore del tronco ad una anteriore.
- *Terminal swing (Deceleration)*, dal 85% al 100% del ciclo del passo: si ha la decelerazione della gamba, la completa estensione del ginocchio e dell'anca, termina quando il piede poggia il terreno ed è in posizione per iniziare un nuovo ciclo del passo.

Inoltre, il passaggio tra la fase di swing e la successiva fase di stance è definita '*heel-strike*' (HS), mentre il passaggio tra la fase di stance e la successiva fase di swing è definita '*toe-off*' (TF).

La **Figura 2** [34] mostra graficamente le fasi del ciclo del passo appena descritte.



*Figura 2. Fasi del ciclo del passo.*

## 1.3 Individuazione delle fasi del ciclo del passo

Il cammino umano è un movimento periodico composto da degli eventi distinti che si susseguono, il che significa che un ciclo del passo può essere diviso in diverse sottofasi [35].

Il rilevamento accurato e rapido delle fasi dell'andatura è fondamentale al fine di ottenere prestazioni ottimali delle protesi motorizzate degli arti inferiori [33]. Lo sviluppo di algoritmi di rilevamento delle fasi affidabili e funzionali è necessario per migliorare la precisione, la stabilità e la sicurezza dei dispositivi protesici e in altre tecnologie riabilitative.

L'individuazione delle fasi e degli eventi che caratterizzano il ciclo del passo è fondamentale per l'elemento protesico che deve attuare delle risposte in funzione di esse. Gli studi più recenti hanno portato alla luce le potenzialità dei sensori inerziali, come nel caso delle IMU, usando in particolar modo i dati forniti da accelerometri e giroscopi.

I ricercatori, come riporta la letteratura, utilizzano diverse terminologie per identificare le fasi del ciclo del passo. Alcuni preferiscono utilizzare *initial contact* (IC) al posto di essere più specifici, in quanto il contatto iniziale avverrebbe con *heel strike* (HS) o con il *toe strike* (TS). In modo simile, alcuni autori, preferiscono l'utilizzo delle parole *end contact* o *foot off*, al posto di utilizzare termini più specifici come *toe off* (TO) o *heel off* (HO). Altri utilizzano i termini *feet adjacent* (FA) per indicare la fase di *initial swing*, oppure si può trovare il termine *tibia vertical* (TBV) per fare riferimento alla fase di *mid-swing* [28]. TO e HS sono, comunque, i due eventi più utilizzati in letteratura per rappresentare le fasi del ciclo del passo [36].

L'attuale stato dell'arte rispetto alla segmentazione dei segnali riguardanti il cammino umano prevede anche l'utilizzo di metodi euristici che utilizzano soglie e su algoritmi che si

basano su delle regole di implementazione (*rule-base methods*). Altri metodi comprendono invece l'utilizzo di tecniche di machine learning che, attraverso l'intelligenza artificiale, sono in grado di trovare dei pattern e riconoscere le diverse fasi a partire dai dati acquisiti [37], anche se, generalmente, i metodi basati su regole offrono tempi di processamento inferiori [31].

## 1.4 Myo Armband

Il Myo Armband è un dispositivo indossabile della Thalmic Labs Inc., sviluppato nel 2014. È dotato di otto sonde elettromiografiche (EMG), un'unità di misura inerziale (IMU), costituita da un giroscopio a 3 assi, accelerometro a 3 assi e magnetometro a 3 assi e un modulo di trasmissione. Il dispositivo è costituito da otto segmenti, collegati tramite un materiale elastico grazie al quale è possibile regolare il bracciale rispetto alle dimensioni dell'arto dell'utente.

Gli elettrodi EMG rilevano i segnali relativi all'attività muscolare e l'IMU rileva i movimenti nello spazio tridimensionale. I dati acquisiti vengono inviati, tramite Bluetooth Low Energy (BLE) integrato nel bracciale, ad altri dispositivi elettronici (attuatori, microcontrollori, ecc.), che svolgono funzioni specifiche a seconda dei dati ricevuti e del software installato.

Il Myo Armband ha una frequenza del segnale di 200 Hz e richiede che l'utente lo indossi e lo sincronizzi con i movimenti dell'arto prima di poter essere utilizzato. Un importante campo di applicazione del bracciale Myo è quello delle protesi; infatti, i sensori mioelettrici

incorporati nel bracciale possono essere sfruttati per rilevare l'attività muscolare degli amputati dell'arto superiore o inferiore e quindi per controllare i movimenti della protesi.

Il bracciale presenta numerosi vantaggi rispetto ad altri dispositivi utilizzati per acquisire l'attività muscolare. Infatti integra, oltre agli elettrodi EMG, anche l'unità IMU che risulta utile a rilevare la posizione dell'arto nello spazio tridimensionale [38].

È presente anche un trasduttore di vibrazione in grado di fornire all'utente un feedback vibratorio a diversi intervalli; in particolare, è possibile dare molte informazioni semplicemente variando la durata della vibrazione, per esempio in caso di avviso di batteria scarica o corretta sincronizzazione del bracciale.

Questo tipo di dispositivo è quindi una soluzione semplice per l'utente che ha la sola esigenza di indossare il bracciale e di non posizionare i sensori EMG sulla pelle con conseguente introduzione di disturbi elettrici che possono deteriorare la rilevazione dei segnali. Myo Armband risulta essere un dispositivo compatto con sensori integrati, unità di elaborazione e trasmissione, che consente la corretta rilevazione dei segnali necessari semplicemente indossandolo, senza l'utilizzo di cavi perché la trasmissione avviene tramite tecnologia BLE.

## **1.5 Unità di misura inerziale**

L'IMU è un sensore che combina accelerometri, giroscopi e magnetometri in grado di misurare, rispetto al corpo su cui viene applicato, l'accelerazione lineare, l'accelerazione angolare e la posizione rispetto al polo nord magnetico in relazione al campo magnetico

terrestre. In particolare, le componenti sono triassiali, in maniera tale da poter fare un'analisi nelle tre direzioni del sistema di riferimento.

È considerato il metodo di misura più efficace nell'analisi del ciclo del passo e nell'individuazione delle fasi di cui esso si compone. Inoltre, in letteratura, si è dimostrato che il posizionamento del sensore sul piede o sulla gamba è più efficace al fine di valutare le fasi del ciclo del passo, rispetto alle IMU posizionate sulla coscia o sull'anca [33], [36].

I sensori inerziali sono utilizzati sempre più frequentemente nello studio della locomozione e sono applicati al monitoraggio di patologie che si riscontrano nella deambulazione, per il rilevamento della stabilità del soggetto e della traiettoria del piede umano, e in generale come approccio di ricerca e di analisi del movimento.

L'unità di misura inerziale è composta dalle seguenti tre parti [9]:

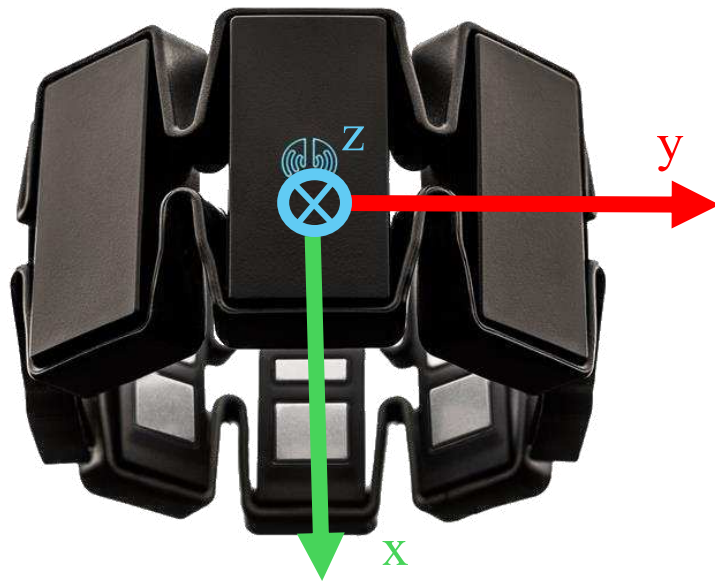
- L'accelerometro, un sensore inerziale che è in grado di misurare l'accelerazione rispetto ai suoi assi. Sfrutta il funzionamento di un elemento meccanico sensibile che comprime una massa di prova attaccata ad un sistema di sospensione. La massa di prova può essere forzata a deviare dalla sua posizione originale grazie all'accelerazione, che può essere misurata elettricamente proporzionalmente agli spostamenti della massa stessa rispetto al sistema di riferimento. Attaccando il sistema sul piede o sulla gamba, si può determinare l'accelerazione dell'arto durante un'analisi del ciclo del passo. Integrando l'accelerazione si ottiene la velocità e integrando nuovamente quest'ultima, si ottiene la posizione.
- Il giroscopio, che misura la velocità angolare. È basato sul concetto di misura della forza di Coriolis, che è una forza apparente proporzionale alla velocità angolare di rotazione rispetto al sistema di riferimento. Come nell'accelerometro, è presente una

massa di prova che cambia la sua posizione iniziale e provoca una forza proporzionale alla velocità angolare e allo spostamento. La forza di Coriolis è rilevata da un sensore capacitivo che la traduce in un segnale proporzionale alla velocità di rotazione. Integrando l'accelerazione angolare possiamo ottenere l'angolo di flessione. Il giroscopio può essere applicato per la misura della postura e del movimento dell'arto su cui viene applicato durante l'analisi del cammino spesso in combinazione con i dati forniti dall'accelerometro, ottenendo così l'accelerazione e la velocità angolare. Ponendolo su piedi o gambe, può essere utilizzato per determinare le varie fasi che compongono il ciclo del passo.

- Il magnetometro, che utilizza l'effetto magnetoresistivo, si basa sulla variazione della resistività di un materiale ferromagnetico percorso da corrente, in cui la variazione della resistenza è proporzionale all'angolo d'inclinazione rispetto alla direzione del campo magnetico. Questi sensori possono stimare i cambiamenti di orientazione dell'arto rispetto al polo Nord magnetico terrestre durante un'analisi del ciclo del passo.

Nel Myo Armband, grazie all'IMU, si possono determinare l'orientazione e i movimenti dell'arto. L'orientazione viene fornita in termini di 'roll', 'pitch' e 'yaw', che rappresentano le possibili rotazioni rispetto agli assi del sistema di riferimento cartesiano. È comunque necessario esprimere le informazioni dell'IMU lungo gli assi anatomici per far sì che esse abbiano un significato anatomico. Sebbene esistano delle procedure di calibrazione utili ad allineare gli assi del sensore inerziale a quelli anatomici [39], esse richiedono elaborazioni che non vengono spesso eseguite. Ci si limita quindi ad allineare manualmente gli assi dell'IMU ad assi ritenuti anatomici. In **Figura 3** [12] vengono descritte le direzioni degli

assi del sensore, in particolar modo il vettore  $y$  rappresenta l'asse trasversale dove avvengono i movimenti di flessione ed estensione, il vettore  $x$  è l'asse longitudinale dove avvengono i movimenti di rotazione e torsione mentre il vettore  $z$  è l'asse antero-posteriore o sagittale dove avvengono i movimenti di allontanamento o avvicinamento della gamba rispetto all'asse longitudinale mediale [40], [41].



*Figura 3. Assi dell'IMU nel Myo Armband.*

Attualmente, il Myo Armband è in grado di estrarre dati IMU ad una frequenza di campionamento di 50 Hz.

## 1.6 Elettromiografia

I segnali mioelettrici sono prodotti da variazioni di potenziale elettrico delle fibre muscolari e possono essere acquisiti da degli elettrodi [42]. Il bracciale Myo utilizza elettrodi di superficie non invasivi, facilmente controllabili, che forniscono informazioni sull'attività muscolare dell'arto ad una frequenza di campionamento di 200 Hz. Il segnale EMG prodotto sotto il controllo del sistema nervoso centrale contiene informazioni sui movimenti e le azioni eseguite dalla gamba, poiché consente di distinguere diversi pattern in relazione al tipo di movimento effettuato [43]. Questo tipo di segnale risulta utile per l'identificazione delle fasi che compongono il ciclo del passo in quanto l'attività muscolare degli arti inferiori si ripete in modo regolare durante il cammino dell'uomo [20].

I segnali EMG grezzi non sono immediatamente utilizzabili, per questo devono essere elaborati al fine di poter estrarre le caratteristiche necessarie alla valutazione dell'attività muscolare e per renderli confrontabili tra più ripetizioni ma anche rispetto a più soggetti. È necessario, dunque, effettuare un pre-processamento dei dati acquisiti. I segnali acquisiti dal dispositivo devono, infatti, prima di tutto essere filtrati e amplificati [7]. Questo perché, molto spesso, risultano essere molto deboli e contaminati da rumore. Il segnale EMG grezzo viene elaborato e segmentato in una serie di finestre di lunghezza finite e, da ognuna di queste, vengono estratte le features. L'estrazione delle features gioca un ruolo principale per raggiungere più alte prestazioni di classificazione utili per il riconoscimento di pattern motori. Generalmente le features utilizzate per il segnale EMG possono essere divise in tre categorie, che includono features nel dominio del tempo (TD), nel dominio della frequenza (FD) e nel dominio tempo-frequenza (TFD). Rispetto alle prime, si valuta l'ampiezza del

segnale, che dipende da quale muscolo si attiva e dalla potenza del segnale, rispetto a come varia nel tempo. Solitamente, ci si focalizza su quest'ultime che sono più semplici a livello computazionale e non richiedono trasformazioni aggiuntive. Nel segnale EMG l'estrazione delle features è fatta per ogni finestra temporale mobile [44]. La grandezza della finestra e la sovrapposizione può variare, per esempio, per quanto riguarda l'utilizzo del Myo Armband, sono state svolte prove con finestre di grandezza 100 ms e sovrapposizione del 50% [45], oppure di 150 ms con overlap del 50% [46], o di 250 ms con un overlap del 50% [47]. Una finestra grande incrementa l'accuratezza della classificazione, in quanto fornisce più dati riguardanti l'EMG. Però si è dimostrato che il ritardo di classificazione deve essere inferiore ai 300 ms, in quanto finestre troppo lunghe ridurrebbero la reattività in tempo reale [48]. Una corta finestra temporale, invece, diminuisce l'accuratezza della classificazione, in quanto aumenta la varianza delle features. In letteratura si evince che, dopo alcuni studi svolti, la lunghezza della finestra non deve essere minore di 125 ms [46].

Un classificatore poi è in grado di riconoscere i pattern del segnale e di distinguerli in diverse classi predefinite. I classificatori sono infatti usati per distinguere le diverse categorie di appartenenza delle features che sono state estratte, inoltre, da studi svolti che andavano a comparare i diversi modelli di classificazione, si è dimostrato che in generale SVM è quello che riporta risultati più performanti [46]. Poiché la classificazione si basa sulle features, che vanno quindi a descrivere le caratteristiche più rilevanti del segnale, è importante andare a fare una giusta selezione di quest'ultime [49].

# Capitolo 2

## Metodi

### 2.1 Partecipanti

Nello studio condotto sono stati presi in esame 20 soggetti, di cui 12 femmine e 8 maschi, nessuno dei quali presenta patologie che coinvolgono gli arti inferiori. Per preservare e proteggere l'identità dei soggetti, ci si riferisce ad essi come Sub01 – Sub20. L'età media dei partecipanti è di 21,3 anni (con un minimo di 17 anni ed un massimo di 30), il peso medio è di 65,8 kg, mentre l'altezza media è 1,71 m. Vengono riportare anche le misure della coscia (dalla testa del femore all'epicondilo laterale) e della gamba (dall'epicondilo laterale al malleolo) che sono rispettivamente in media pari a 466,45 cm e 372,6 cm. In aggiunta, 16 persone hanno come gamba dominante la destra, mentre 6 la sinistra. La gamba dominante è quella con cui il soggetto si trova più a suo agio nell'iniziare ogni prova del test.

Le informazioni relative ai soggetti sono contenute in un file chiamato "SubjectInformation.xlsx" che presenta l'ID del partecipante, l'età, il genere (se maschio o femmina), l'altezza, il peso, la gamba con la quale inizia ogni prova, la lunghezza della coscia e della gamba. Una descrizione più dettagliata è rappresentata nella **Tabella 1** [50] che contiene tutte le caratteristiche relative ad ogni soggetto.

Sub	Età	Genere	Altezza [m]	Peso [kg]	Gamba dominante	Coscia [mm]	Gamba [mm]
Sub01	30	M	1.78	80	L	460	390
Sub02	17	F	1.63	55	R	455	350
Sub03	17	F	1.58	55	R	410	325
Sub04	18	F	1.60	51	R	460	330
Sub05	22	F	1.63	60	R	420	330
Sub06	21	F	1.70	60	R	460	370
Sub07	21	M	1.87	83	R	530	430
Sub08	22	M	1.78	75	R	469	445
Sub09	22	M	1.81	63	L	450	400
Sub10	22	M	1.80	67	R	500	380
Sub11	21	F	1.48	46	R	440	315
Sub12	21	F	1.65	58	R	440	432
Sub13	22	F	1.70	60	L	450	360
Sub14	21	F	1.80	65	R	510	390
Sub15	21	F	1.70	70	R	490	395
Sub16	21	F	1.75	75	R	440	360
Sub17	21	M	1.80	78	L	530	410
Sub18	24	M	1.87	83	R	555	430
Sub19	22	M	1.71	78	R	415	350
Sub20	20	F	1.65	54	R	445	350

*Tabella 1. Caratteristiche dei partecipanti.*

## 2.2 Protocollo di misura

Per acquisire i dati necessari che andranno successivamente ad essere organizzati in un dataset, sono state eseguite, per ogni soggetto, prove di cammino rispetto a tre diversi tipi di condizioni di locomozione.

Ogni partecipante per la durata di tutte le prove ha indossato il dispositivo Myo Armband posizionato approssimativamente a metà gamba ed allineato rispetto alla lunghezza della tibia (**Figura 4** [51], [52]), con una frequenza di acquisizione di 200 Hz per l'EMG e di 50 Hz per l'IMU. Il bracciale mioelettrico, per ogni test, è rimasto sempre nello stesso punto, mantenendo quindi così una coerenza costante nella posizione rispetto a quella originaria.



*Figura 4. Visualizzazione grafica del dispositivo indossato sulla gamba.*

Il primo test è stato eseguito in piano, facendo camminare il soggetto senza alcun vincolo di velocità in un corridoio lungo 15 m. Il soggetto doveva partire da fermo, percorrere il corridoio fino alla fine, attendere circa due secondi prima di girarsi e, dopo essersi voltato, attendere nuovamente lo stesso intervallo di tempo per poi tornare al punto di partenza. L'acquisizione, per tutte le prove, iniziava prima della partenza del soggetto e terminava una volta che esso si fermava al punto di arrivo. Inoltre, sia durante la partenza, che dopo aver effettuato il cambio di direzione, il soggetto era tenuto a partire con la propria gamba

dominante. Dopodiché, il partecipante ritornava nella posizione di partenza e ricominciava nuovamente la camminata per un totale di cinque prove. Il test veniva ripetuto qualora il soggetto non rispettasse le istruzioni, come nel caso in cui non attendesse l'intervallo di tempo dopo il cambio di direzione o iniziasse a camminare con la gamba sbagliata.

Con le stesse metodologie e accorgimenti sono state eseguite le prove su un piano inclinato, dove sono state eseguite cinque prove per ogni utente, con le stesse procedure utilizzate rispetto alla camminata in piano.

Le ultime prove, invece, riguardavano la salita e la discesa di 9 scalini. Il soggetto doveva partire da fermo e, dopo l'avvio dell'acquisizione, iniziare la salita delle scale, attendere circa due secondi prima di girarsi e altrettanto dopo essersi girato, per poi iniziare la discesa. Ogni prova prevedeva due esecuzioni consecutive, quindi con un totale di due salite e due discese.

In **Figura 5** vengono riportati i segnali grezzi del giroscopio registrati durante le diverse modalità di cammino (camminata in piano, su rampa e salita/discesa di scale) per tutta la durata di una prova, eseguita nelle modalità descritte precedentemente. La **Figura 6** mostra, invece, un dettaglio di alcuni cicli del passo relativi alla stessa prova.

I segnali elettromiografici grezzi relativi a ciascun elettrodo sono riportati in **Figura 7** per il cammino in piano, in **Figura 8** per la superficie inclinata e in **Figura 9** per le scale.

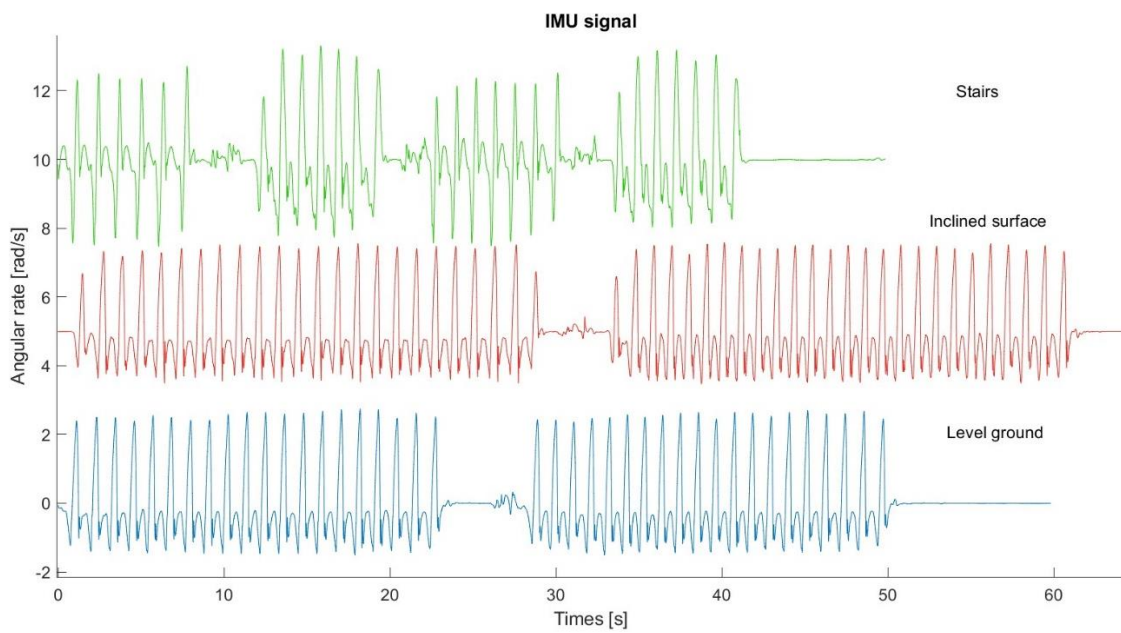


Figura 5. Sub01, velocità angolare acquisita nelle tre prove (camminata in piano, su rampa, salita e discesa di scale).

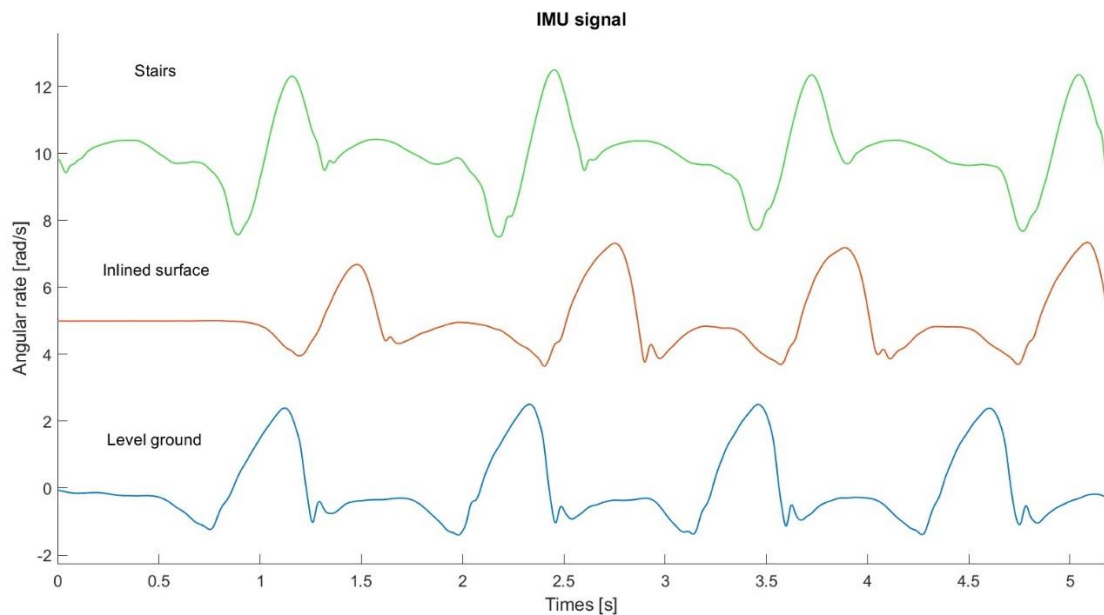


Figura 6. Sub01, dettaglio dei primi cicli del passo della velocità angolare acquisita nelle tre prove (camminata in piano, su rampa, salita e discesa di scale).

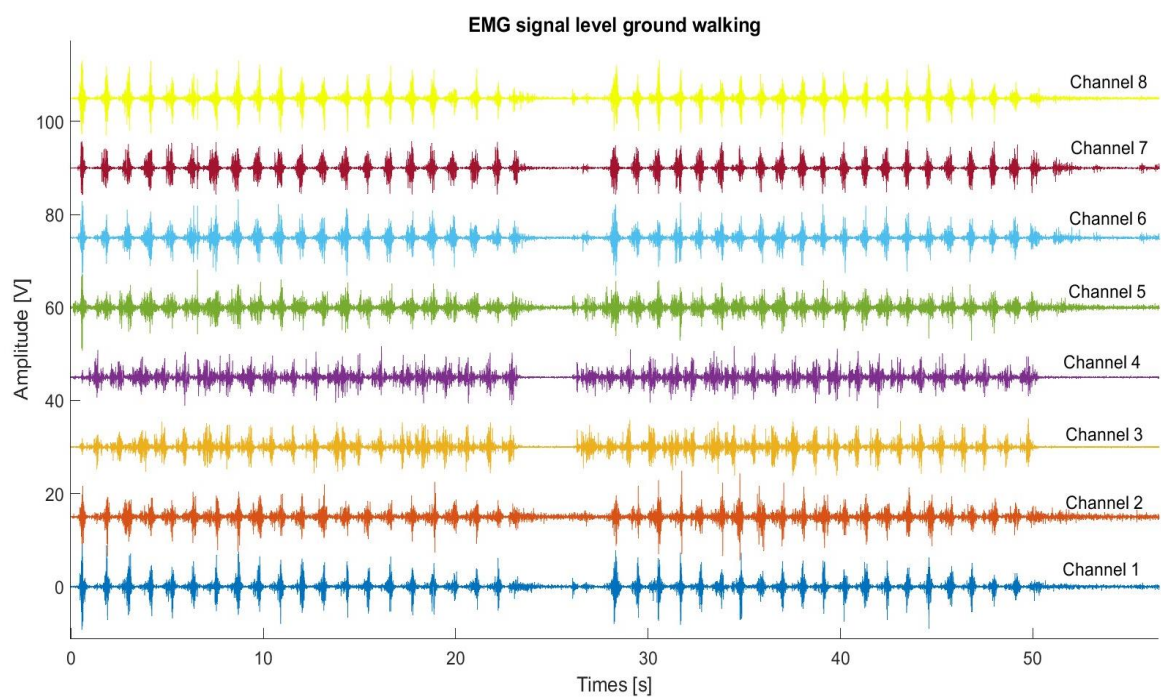


Figura 7. Sub01, EMG acquisito durante la camminata in piano.

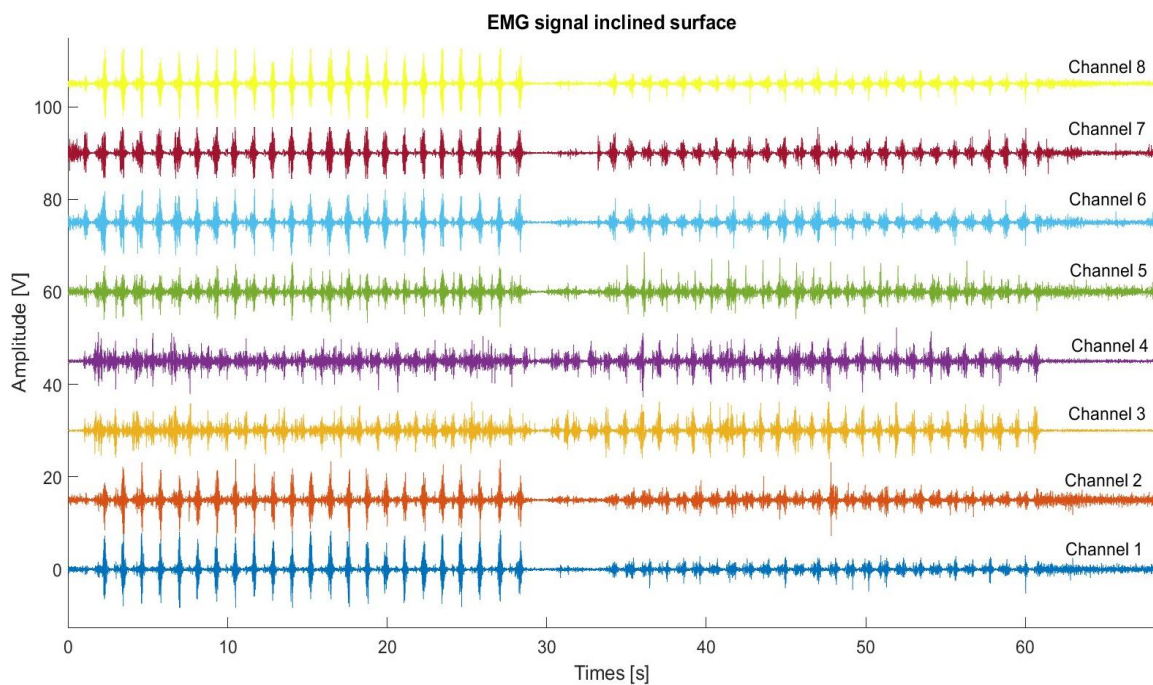
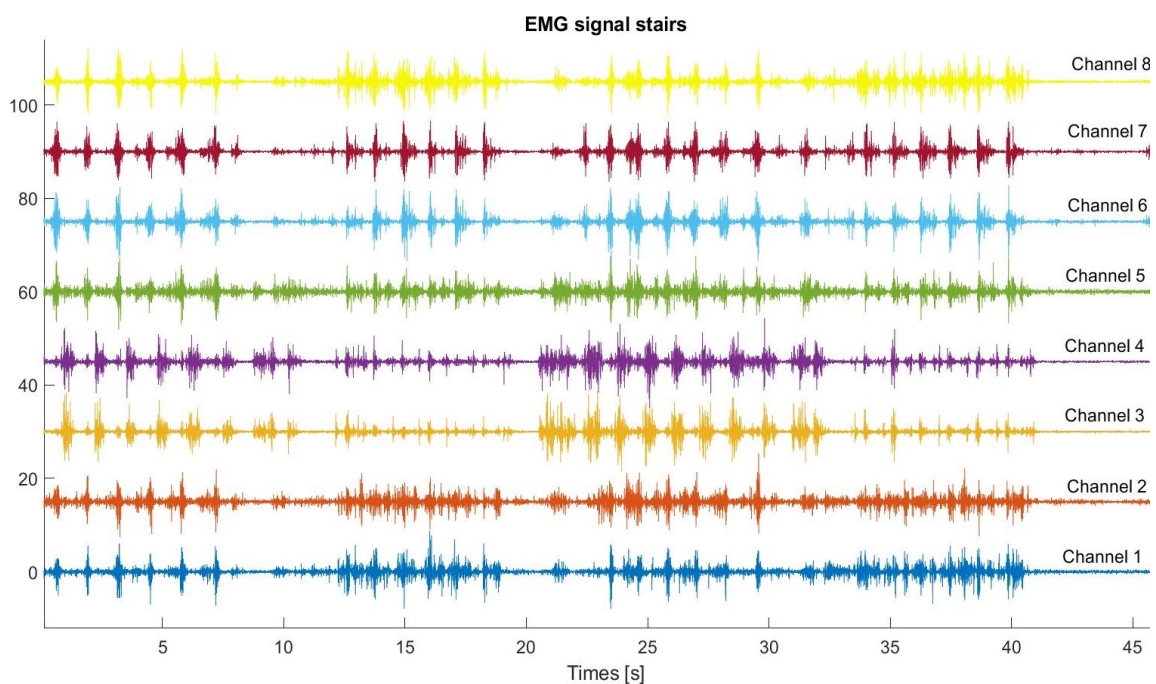


Figura 8. Sub01, EMG acquisito durante la camminata su superficie inclinata.



*Figura 9. Sub01, EMG acquisito durante salita e discesa di scale.*

## 2.3 Elaborazione dei dati

I dati relativi a ciascuna prova, una volta acquisiti, sono stati salvati in una cartella, contenente i segnali grezzi di ciascun soggetto. Sono poi stati elaborati attraverso il software Matlab R2023b [53] al fine di effettuare delle analisi successive.

In primo luogo, poiché i dati dell'IMU e dell'EMG sono campionati a frequenze diverse, è stato effettuato un ricampionamento per uniformare la frequenza di campionamento di entrambi i segnali a 200 Hz. Inoltre, sono state sincronizzate le lunghezze dell'IMU e

dell'EMG, in modo da ottenere segnali della stessa durata temporale. Per le analisi successive, volte ad individuare le varie fasi del ciclo del passo, si è usato il segnale dell'asse y del giroscopio.

I dati provenienti dalle prove eseguite in piano non sono stati filtrati poiché privi di rumori significativi che potessero compromettere l'analisi successiva relativa alla segmentazione.

Per i segnali acquisiti durante le prove eseguite su un piano inclinato, è stato applicato un filtro Butterworth passa-basso del secondo ordine con una frequenza di taglio di 5 Hz. Mentre, per quanto riguarda le prove eseguite sulle scale, i segnali sono stati filtrati con un filtro Butterworth del secondo ordine, utilizzando una frequenza di taglio di 10 Hz. Questa scelta è stata fatta per minimizzare il ritardo di individuazione delle varie fasi del ciclo del passo, garantendo una stretta somiglianza con il segnale grezzo originale [30].

La segmentazione del segnale nelle relative fasi è stata resa efficace grazie all'uso di soglie temporali e di ampiezza, che hanno permesso di isolare il segnale relativo esclusivamente alla camminata del soggetto, escludendo i momenti in cui avvenivano cambi di direzione. Inoltre, sono stati esclusi il primo e l'ultimo passo di ogni prova, poiché rappresentavano momenti di accelerazione e decelerazione, in cui il rumore sovrapposto al segnale era più evidente e non rifletteva la normale deambulazione del soggetto. Per lo stesso motivo, non sono stati considerati i cicli del passo immediatamente precedenti e successivi ad ogni cambio di direzione.

L'individuazione delle varie fasi del ciclo del passo è stata eseguita utilizzando algoritmi differenti, a seconda che la prova fosse condotta su un piano, su una rampa o sulle scale.

La segmentazione è stata utile per creare etichette specifiche per ciascun soggetto, consentendo di monitorare l'evoluzione nel tempo delle diverse fasi del ciclo del passo durante il cammino di ogni individuo, contribuendo così al completamento del dataset finale.

## 2.4 Rule-based methods

Il metodo più utilizzato per rilevare le diverse fasi del ciclo del passo consiste nell'utilizzo di algoritmi basati su regole computazionali (*rule-base methods*), che permettono di suddividere il segnale proveniente dall'IMU in vari eventi. Il largo utilizzo di questi metodi è dettato dal fatto che sono semplici, intuitivi e richiedono meno complessità computazionale, che porta quindi ad avere una minore latenza nei processi in cui sono coinvolti [36].

Un modo per implementare tale metodo è quello di impostare delle soglie sul segnale grezzo o elaborato. Spesso si utilizzano più regole costruite su dichiarazioni condizionali (logica *if-else*) che sono collegate mediante vincoli di disuguaglianza o operatori logici (AND/OR). Molto spesso si utilizzano algoritmi di zero-crossing uniti a tecniche di rilevamento di picchi (*peak detection*) che possono indicare dei precisi eventi relativi al segnale [36], [54], [24].

Attraverso questi metodi, le misurazioni effettuate attraverso il giroscopio sono la migliore scelta per la segmentazione del ciclo del passo, in quanto il segnale contiene dei pattern ben distinguibili [55]. Infatti, rispetto agli studi eseguiti in letteratura, i punti TO (*Toe Off*) e IC (*Initial Contact*) corrispondono ai due picchi negativi che avvengono prima e dopo un

massimo picco conosciuto come MS (*Mid-swing*) rispetto alla velocità angolare acquisita ponendo il sensore sulla gamba [24], [10].

Un'alternativa utilizzata è quella che prevede l'utilizzo di metodi di machine learning che, però, richiedono grandi campioni di dati provenienti da molti partecipanti e risulta essere anche più costosa a livello computazionale [55], [24].

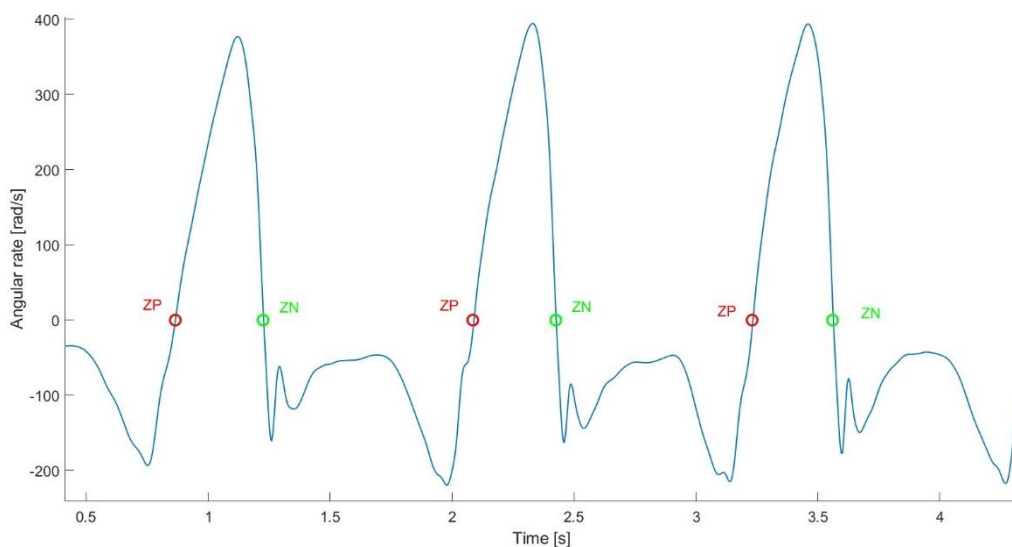
Altri eventi relativi al ciclo del passo sono rilevati attraverso differenti metodi, basati su regole e algoritmi progettati per segmentare il segnale nelle diverse fasi utilizzando per lo più le informazioni fornite dall'unità di misura inerziale, in particolare dal giroscopio, che mostra pattern definiti e ripetibili.

## 2.5 Camminata in piano

Per la segmentazione dei segnali relativi alle prove svolte in piano, ci si è basati sullo studio condotto da Salminen et al. (2024) [28], al fine di individuare le fasi di *initial contact* (IC), *heel rise* (HR), *toe off* (TO), *feet adjacent* (FA) e *tibia vertical* (TBV).

Nel suddetto studio, IC rappresenta l'istante in cui il tallone tocca il suolo per la prima volta, segnando l'inizio del ciclo del passo; HR è il punto in cui il tallone si solleva da terra; TO indica il momento in cui il piede si stacca completamente dal terreno; FA corrisponde all'istante in cui il piede in oscillazione si avvicina a quello in appoggio; infine, TBV

rappresenta il momento in cui la tibia raggiunge la posizione verticale durante la fase di swing (mid-swing). Il ciclo finisce quando si ritorna al punto di IC. Questi eventi vengono determinati a partire dalla componente medio-laterale della velocità angolare del segnale, individuando i punti in cui il segnale attraversa lo zero (*zero-crossing*). In particolare, si fa riferimento allo zero-crossing (**Figura 10**) in cui il segnale ha una pendenza negativa come ZN, mentre quando la pendenza è positiva è denominato ZP.



*Figura 10. Individuazione dei punti di zero-crossing.*

Il punto IC, come si può notare in **Figura 11**, viene individuato come il primo minimo locale dopo un ZN del segnale non filtrato.

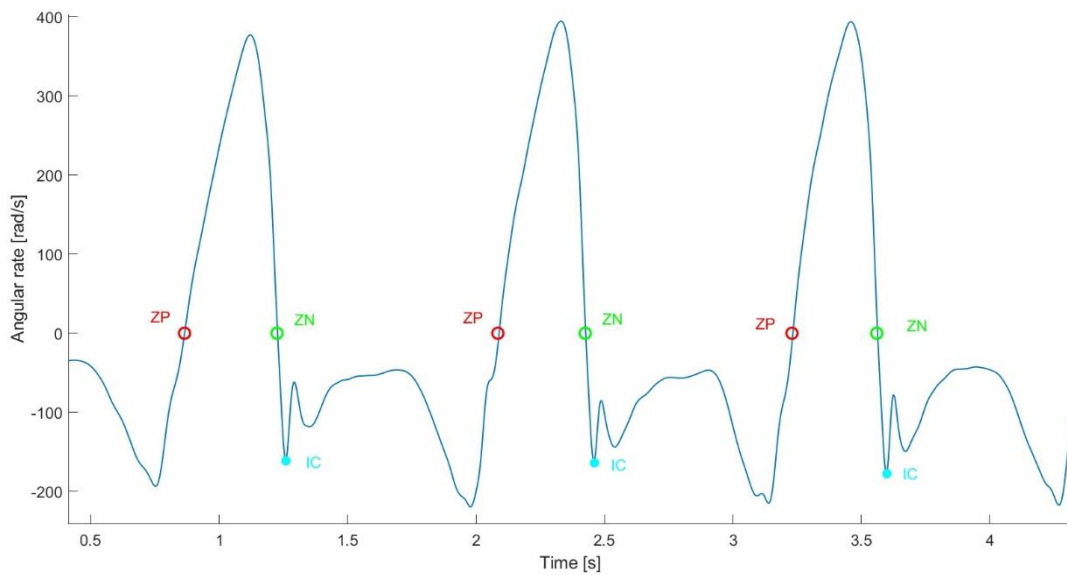


Figura 11. Individuazione dei punti IC.

Per gli altri eventi, invece, si fa riferimento invece al metodo del calcolo dell'area cumulativa sopra e sotto il segnale relativo alla componente medio laterale della velocità angolare (CSAV) [28]. In particolare, si fa riferimento all'area cumulativa negativa della velocità angolare tra uno ZN e il successivo ZP durante la fase di stance ( $CSAV_{ST}$ ) e all'area cumulativa positiva della velocità angolare tra uno ZP e il successivo ZN durante la fase di swing ( $CSAV_{SW}$ ).

Gli altri eventi sono stati trovati tramite test eseguiti con pedane di forza e sistemi di motion capture che hanno permesso di correlare i punti relativi alle diverse fasi del ciclo del passo con il segnale, associando i punti a diverse percentuali delle aree cumulative del segnale stesso. La valutazione della posizione degli eventi che occorrono durante il ciclo del passo effettuata per il tramite di pedane e marker ottici, è considerata come il “gold standard” per la misurazione dei parametri temporali del cammino. Le percentuali sono state fornite dagli

autori dell'articolo stesso basandosi su un modello statistico da loro riportato. Gli eventi HR e TO si sono verificati durante la fase  $CSAV_{ST}$ , mentre FA e TBV sono stati inclusi nella fase  $CSAV_{SW}$ . In particolare, gli eventi HR e TO, si trovano rispettivamente al 46,0% e al 95,7% della  $CSAV_{ST}$ , mentre FA e TBV sono stati determinati rispettivamente al 20,0% e al 73,1% della  $CSAV_{SW}$ . I valori sono stati aggiustati con fattori correttivi trovati mediante analisi statistiche. In **Figura 12** [28] è rappresentata graficamente la descrizione del metodo utilizzato per individuare le fasi del ciclo del passo mediante le prove effettuate in piano.

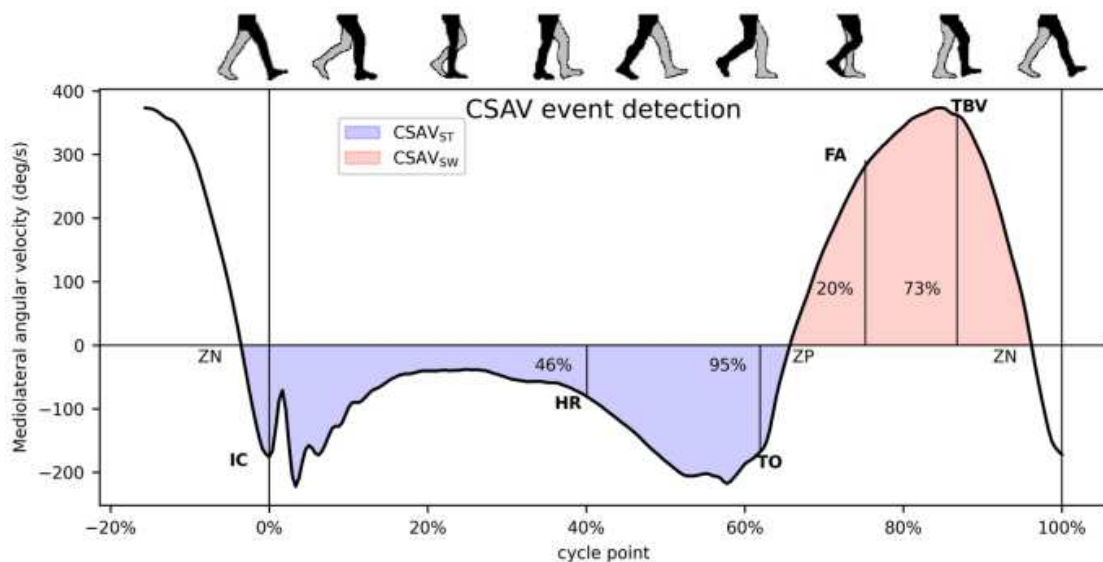


Figura 12. Velocità angolare medio laterale di un tipico ciclo del passo, in cui si evidenziano gli eventi IC, HR, TO, FA, TBV.

## 2.6 Camminata su rampa

L'individuazione degli eventi relativi al cammino rispetto alle prove eseguite su un piano inclinato è stata fatta a partire dallo studio eseguito da Gouwanda e Gopalai (2015) [23]. Ciò ha permesso di individuare, a partire dalla componente medio laterale del segnale filtrato relativo alla velocità angolare, le fasi di *heel strike* (HS), *toe off* (TO) che rappresentano corrispettivamente l'inizio della fase di stance e di swing e il punto di *mid-swing* (MS).

Anche in questo metodo si individuano i punti in cui il segnale passa per lo zero, e successivamente si trovano i punti di MS, HS e TO. In particolare, grazie all'uso di soglie stabilite sperimentalmente, che consentono di individuare gli eventi con maggiore precisione, MS viene identificato come il punto in cui, superata una certa soglia, la derivata prima del segnale passa da positiva a negativa. Una volta individuato MS, HS viene determinato come il minimo locale successivo, mentre TO viene identificato come il minimo locale che precede il punto MS successivo. In **Figura 13** [23] viene rappresentato graficamente il metodo descritto che ha permesso l'individuazione delle fasi di MS, HS, TO per le prove eseguite su un piano inclinato.

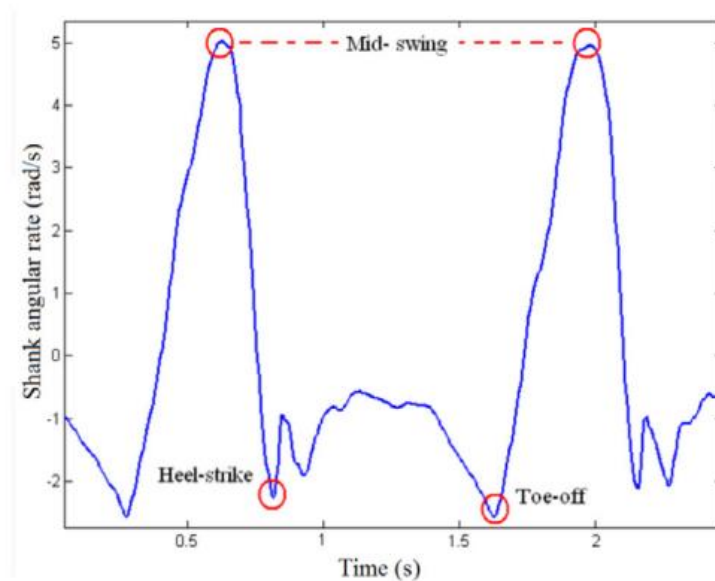
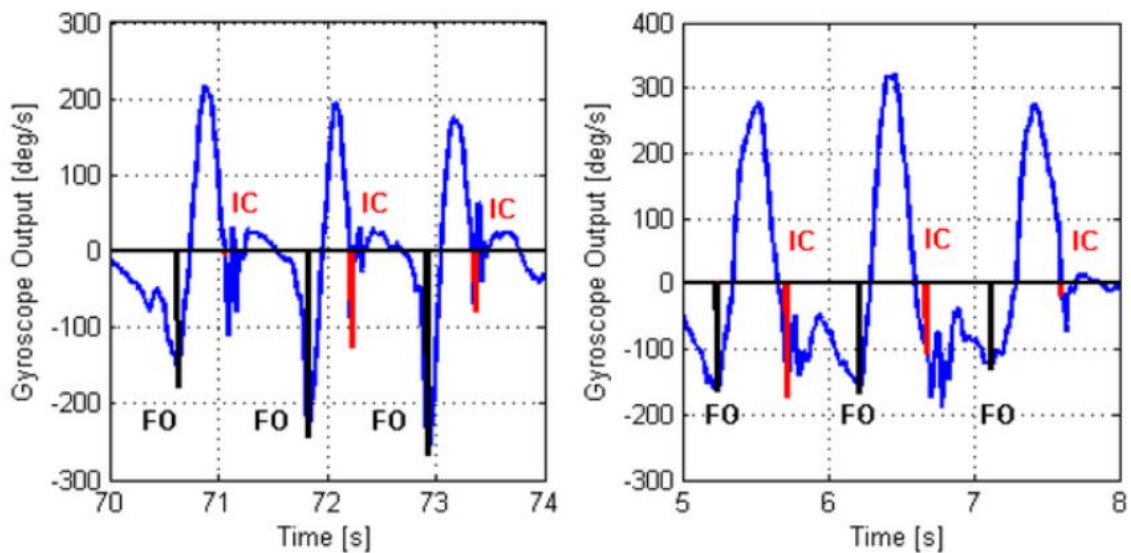


Figura 13. Componente medio laterale della velocità angolare durante il cammino su superficie inclinata e individuazione delle fasi di MS, HS, TO.

## 2.7 Salita e discesa di scale

Rispetto all'individuazione delle fasi del ciclo del passo relative alle prove eseguite sulle scale, si è preso come riferimento lo studio condotto da Formento et al. (2014) [56]. Grazie al metodo descritto nell'articolo di riferimento, si sono potute individuare le fasi di *initial contact* (IC) e di *foot off* (FO). Anche in questo metodo si determinano, inizialmente, i punti in cui il segnale attraversa lo zero, sia in senso discendente che ascendente. Il primo minimo locale trovato dopo ogni zero-crossing discendente viene identificato come IC. Dopo aver individuato IC, viene stabilito un tempo di attesa, al termine del quale il picco negativo

successivo rappresenta il punto FO. Una volta determinato FO, l'algoritmo riparte da capo. La **Figura 14** [56] illustra graficamente l'individuazione dei punti nel segnale registrato rispetto alla salita e alla discesa delle scale. Si osserva, inoltre, che rispetto ai segnali precedentemente acquisiti in condizioni di cammino su superficie piana (**Figura 12**) e inclinata (**Figura 13**), i grafici relativi alle prove eseguite su scala presentano un maggior livello di rumore. Di conseguenza, per ottenere una segmentazione più accurata e una migliore individuazione delle fasi e degli eventi del cammino, in questo tipo di segnali si preferisce prendere in considerazione un numero inferiore di punti.



*Figura 14. Segnale relativo al giroscopio durante salita (figura a sinistra) e discesa (figura a destra) di scale con i relativi punti IC e FO.*

## 2.8 Elaborazione del segnale elettromiografico

Il segnale elettromiografico grezzo è stato filtrato con un filtro Butterworth passa alto del secondo ordine con una frequenza di taglio di 10 Hz per eliminare il rumore a bassa frequenza che potrebbe interferire con il segnale. Inoltre, è stata eseguita l'estrazione delle features per ogni finestra temporale mobile di 200 ms con un incremento di 40 ms. Le features che sono state estratte dal segnale EMG e che dalla letteratura presentano un più alto grado di classificazione, soprattutto quando si tratta di dispositivi indossabili che lavorano a più basse frequenze [57], sono le seguenti:

- Waveform Length (WL), è una misura della complessità del segnale EMG e rappresenta la lunghezza cumulativa della forma d'onda del segnale rispetto al tempo.

$$WL = \sum_{k=1}^{N-1} |x_{k+1} - x_k|$$

- Root Mean Square (RMS), è una media quadratica del segnale e rappresenta la sua potenza media.

$$RMS = \sqrt{\frac{1}{N} \sum_{k=1}^N x_k^2}$$

- Zero Crossing (ZC), che valuta quante volte il segnale attraversa lo zero; quindi, quando l'ampiezza diventa positiva o negativa. È necessario applicare una soglia per sopprimere gli attraversamenti dello zero indotti dal rumore. Considerando quindi

una soglia  $T$ , una finestra del segnale contenente  $N$  campioni e guardando due campioni consecutivi  $x_k$  e  $x_{k+1}$ , c'è un passaggio per lo zero tra i due se è soddisfatta la seguente condizione:

$$g_{ZC}(x_k) = \begin{cases} 1 & \text{se } (x_k \cdot x_{k+1}) > 0 \text{ e } |x_k - x_{k+1}| \geq T \\ 0 & \text{altrimenti} \end{cases}$$

- Slope Sign Change (SSC), che valuta quante volte il segnale passa da una pendenza positiva ad una negativa. Per non considerare la variazione di pendenza dovuta dal rumore, si considerano una soglia  $T$  e tre campioni consecutivi  $x_{k-1}$ ,  $x_k$  e  $x_{k+1}$  del segmento del segnale con  $N$  campioni:

$$(x_k - x_{k-1})(x_k - x_{k+1}) \geq T$$

Sono stati applicati poi algoritmi di classificazione per associare le caratteristiche del segnale EMG alle fasi segmentate tramite il sensore inerziale. I due segnali sono stati, infatti, precedentemente sincronizzati nel tempo, per garantire la corretta analisi. La classificazione è stata eseguita attraverso gli algoritmi SVM (Support Vector Machine) e LDA (Linear Discriminant Analysis) per tutte le prove di cammino in piano di ogni soggetto.

## 2.9 Organizzazione del dataset

Al fine di organizzare i dati acquisiti ed elaborati, si è fatto riferimento al modello di dataset, denominato SIAT-LLMD, proposto da Wei et al. (2023) [8] che presenta una struttura che raccoglie dati cinematici, cinetici ed elettromiografici, con le corrispettive etichette. Il dataset proposto dall'articolo di riferimento è stato sviluppato considerando 40 soggetti, ai quali è stato richiesto di effettuare 16 diversi tipi di movimento; i dati cinematici e cinetici sono stati collezionati usando un sistema di motion capture e delle piattaforme di forza, mentre per l'EMG si sono utilizzati nove sensori posizionati sulla coscia e sulla caviglia della gamba sinistra di ogni soggetto. Precisamente, questi compiti includono lo stare in piedi fermi, camminare su un terreno pianeggiante, alzarsi, sedersi, salire e scendere le scale, sollevare il ginocchio, inclinare la punta del piede, sollevare la gamba in avanti, portare la gamba all'indietro, sollevare il ginocchio e in seguito portare il polpaccio in avanti, sollevare la gamba lateralmente, affondo in avanti e affondo all'indietro.

La particolarità di SIAT-LLMD è il fatto che propone etichette volte a classificare i diversi tipi di movimento e le varie fasi del ciclo del passo. Queste sono utili al fine di avere un dataset più facilmente interpretabile e possono essere usate come parametri di riferimento per la validazione dell'algoritmo utilizzato per il partizionamento del ciclo del passo [58].

Il dataset è strutturato in modo tale da contenere, per ogni soggetto, una cartella denominata "Data", contenente i file relativi ad ogni prova in formato Excel. All'interno del file la prima colonna è dedicata al tempo. I dati cinematici sono collocati dalla seconda alla nona colonna, quelli cinetici dalla decima alla diciassettesima colonna, mentre i dati sEMG (elettromiografia di superficie) sono registrati dalla diciottesima alla ventiseiesima colonna.

Inoltre, è presente anche una cartella denominata “Labels” in cui ogni prova di ogni partecipante è descritta da una tabella in formato Excel. Nel file, la prima colonna contiene il vettore tempo, la seconda riporta le etichette che indicano quando avvengono i cambiamenti delle varie fasi che compongono il ciclo del passo, mentre la terza colonna rappresenta il gruppo di movimento effettuato.

Il primo tipo di etichetta include “A” e “R”, corrispondenti allo stato attivo o a riposo del soggetto rispetto ai movimenti degli arti inferiori. Le etichette sono determinate dalle piattaforme di forza e dai dati cinematici. Il secondo tipo di etichetta indica le diverse fasi dell'andatura. Vengono riportati cinque eventi dell'andatura relativi alla camminata in piano che includono *heel strike* (HS), *maximum stance flexion* (MSF), *maximum stance extension* (MSE), *toe-off* (TO) e *maximum swing flexion* (MWF), tre dei quali (HS, TO e MWF) possono essere rilevati anche quando si sale e si scende le scale.

Nello studio condotto nel seguente elaborato, è stato seguito il modello di strutturazione del dataset SIAT-LLMD, con l’obiettivo di utilizzare un set up sperimentale che risulti essere ancora più semplice e facilmente replicabile. Inoltre, gli eventi relativi alla camminata sono stati individuati in modo da essere in linea con le ricerche più recenti presenti in letteratura. In particolar modo, per ogni soggetto, sono state create due cartelle. La prima denominata “Data” contiene i dati acquisiti relativi sia all’IMU che all’EMG, organizzati in una tabella in formato “.xls” con 15 colonne. La prima colonna rappresenta il tempo, dalla seconda alla quarta sono riportati i dati dell’accelerazione lungo gli assi x, y, z, dalla quinta alla settima quelli della velocità angolare rispetto agli stessi assi e dall’ottava alla quindicesima sono inclusi i dati elettromiografici relativi agli otto elettrodi. La seconda cartella chiamata “Label” fa riferimento alle etichette trovate andando a segmentare il ciclo del passo durante

le prove eseguite, in modo tale da poter visualizzare in modo chiaro il cambiamento delle varie fasi rispetto al tempo. Anche in questo caso, i dati sono strutturati in una tabella in formato “.xls” contenente 3 colonne, in cui la prima è relativa al tempo, la seconda associa dei valori numerici per ogni fase trovata, mentre la terza contiene il gruppo di movimento effettuato durante la prova.

All'interno del dataset è inoltre disponibile un file in formato “.xlsx” che sintetizza tutte le informazioni di ogni soggetto che ha partecipato al test.

## 2.10 Implementazione degli algoritmi

Per completare il dataset, sono stati sviluppati i seguenti algoritmi in Matlab, che hanno avuto i seguenti obiettivi:

- SaveData.m salva e organizza i dati acquisiti in una tabella contenente le informazioni provenienti dall'accelerometro, dal giroscopio e dagli 8 canali EMG, e successivamente salva l'intera tabella in un file Excel per tutte le prove di ogni soggetto. La tabella (come descritto in **Figura 15**) contiene nella prima colonna il vettore tempo, dalla seconda alla quarta i dati dell'accelerometro nelle tre componenti spaziali, dalla quinta alla settima i dati del giroscopio nelle tre componenti spaziali e dall'ottava alla quindicesima gli 8 canali EMG.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Column 1: TIME	Column 2-4: Kinematic (ACC)			Column 5-7: Kinematic (GYRO)			Column 8-15: sEMG (8 channels)							
2															
3															
4															
5															
6															
7															
8															
9															
10															

Figura 15. Organizzazione dei dati acquisiti in formato Excel.

- Segmentation\_WAK.m consente di selezionare da tastiera il soggetto e la prova da visualizzare relativa alla camminata in piano. Successivamente, tramite dei plot, viene mostrato il segnale con le relative fasi del ciclo del passo, evidenziate sopra il segnale stesso tramite simboli e colori, insieme ai valori delle label sovrapposti (a gradino). La scelta di progettare un codice che acquisisse da tastiera la prova desiderata per ciascun soggetto è stata presa per evitare l'eccessivo sforzo computazionale richiesto per visualizzare graficamente tutte le prove di tutti i soggetti con la relativa segmentazione.
- Segmentation\_RAMP.m esegue la stessa funzione ma rispetto ai dati relativi alla camminata su piano inclinato.
- Segmentation\_STAIR.m per le prove di salita e discesa eseguite sulle scale.
- SaveLabel\_WAK.m consente di salvare in una tabella le label individuate, come descritto nei capitoli precedenti, relative alle fasi del ciclo del passo per ciascuna

prova di ogni soggetto. I dati vengono salvati in una tabella in formato Excel (come riportato in **Figura 16**), dove la prima colonna contiene il vettore tempo, la seconda colonna riporta le label e la terza indica il gruppo di movimento della prova, che è sempre 1, poiché il protocollo sperimentale prevedeva che il soggetto eseguisse sempre lo stesso movimento per ogni prova.

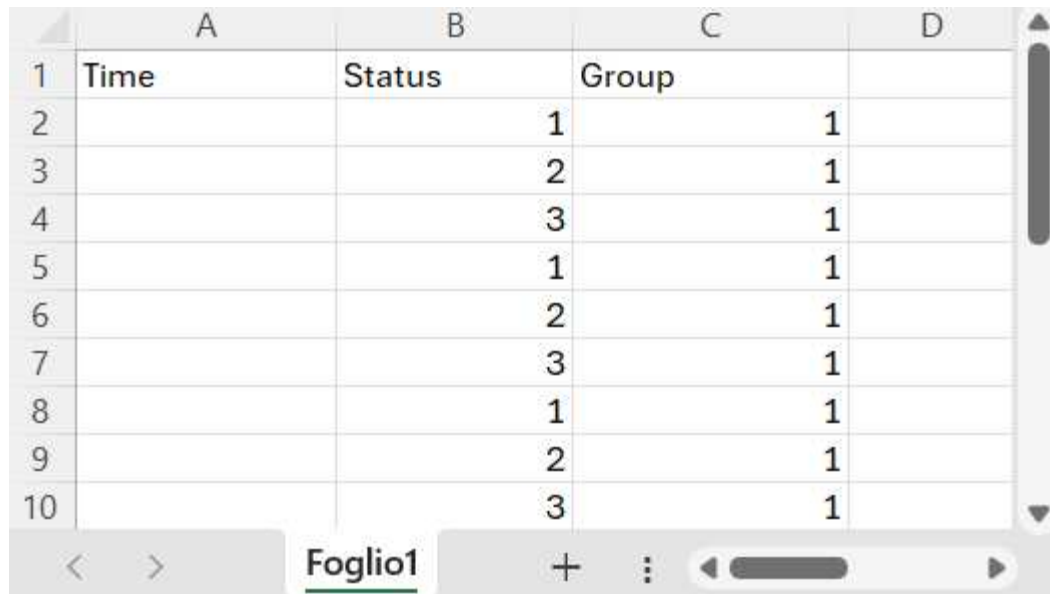
Nella seconda colonna, si assegna il valore 1 per la fase tra IC e HR, 2 tra HR e TO, 3 tra TO e FA, 4 tra FA e TBV, 5 tra TBV e IC; altrimenti, viene assegnato NaN (Not a Number) sia quando il soggetto è fermo sia durante i cambi di direzione.

	A	B	C	D
1	Time	Status	Group	
2		1		1
3		2		1
4		3		1
5		4		1
6		5		1
7		1		1
8		2		1
9		3		1
10		4		1
11		5		1
12		1		1

*Figura 16. Modello di tabella in formato Excel relativa alla segmentazione delle prove eseguite in piano.*

- SaveLabel\_RAMP.m esegue la stessa funzione ma relativa alle prove eseguite su un terreno inclinato. In questo caso la seconda colonna assegna il valore 1 tra HS e TO, 2 tra TO e MS e 3 tra MS e HS; se il soggetto è fermo o sta eseguendo cambi di

direzione, viene assegnato NaN. La **Figura 17** rappresenta un prototipo di tabella che contiene le caratteristiche descritte.



	A	B	C	D
1	Time	Status	Group	
2		1	1	
3		2	1	
4		3	1	
5		1	1	
6		2	1	
7		3	1	
8		1	1	
9		2	1	
10		3	1	

*Figura 17. Modello di tabella in formato Excel relativa alla segmentazione delle prove eseguite su piano inclinato.*

- SaveLabel\_STAIR.m consente di salvare allo stesso modo su una tabella in formato Excel (come rappresentato in **Figura 18**), l'analisi che riguarda la segmentazione rispetto alle prove eseguite sulle scale. In questo caso, nella seconda colonna si assegna il valore 1 tra IC e FO, 2 tra FO e IC e NaN se il soggetto è fermo o sta eseguendo cambi di direzione.

	A	B	C	D
1	Time	Status	Group	
2		1	1	
3		2	1	
4		1	1	
5		2	1	
6		1	1	
7		2	1	
8		1	1	
9		2	1	
10		1	1	

*Figura 18. Modello di tabella in formato Excel relativa alla segmentazione delle prove eseguite su scale.*

- La funzione `findZeroCrossing.m` è stata usata negli algoritmi precedenti per trovare i punti in cui il segnale interseca l'asse x e per specificare, inoltre, se il passaggio per lo zero avviene in senso negativo o positivo.

# Capitolo 3

## Risultati

### 3.1 Creazione del data set

Il dataset finale raccoglie tutte le prove di ogni soggetto in una struttura organizzata e facilmente accessibile. È formato dalle seguenti parti, schematizzate graficamente in **Figura 19** [50]:

- Un file denominato “SubjectInformation.xlsx” contenente le informazioni di ogni soggetto che ha partecipato allo studio (come descritto nel capitolo 2.1).
- Una cartella chiamata “Codes” relativi ai codici sviluppati in Matlab (come riportato nel capitolo 2.10).
- 20 cartelle, una per ogni soggetto che ha partecipato alla creazione del dataset, ciascuna delle quali presenta:
  - una cartella denominata “Data” che racchiude in formato Excel tutte le prove svolte dal soggetto (camminata in piano, su rampa e salita/discesa di scale), organizzando i dati, acquisiti con il Myo Armband, in tabelle (capitolo 2.10).
  - Una cartella chiamata “Label” che fornisce informazioni attraverso tabelle in formato Excel rispetto all’elaborazione dei dati e alla segmentazione del ciclo del passo, per ogni prova svolta dal soggetto (capitolo 2.10).

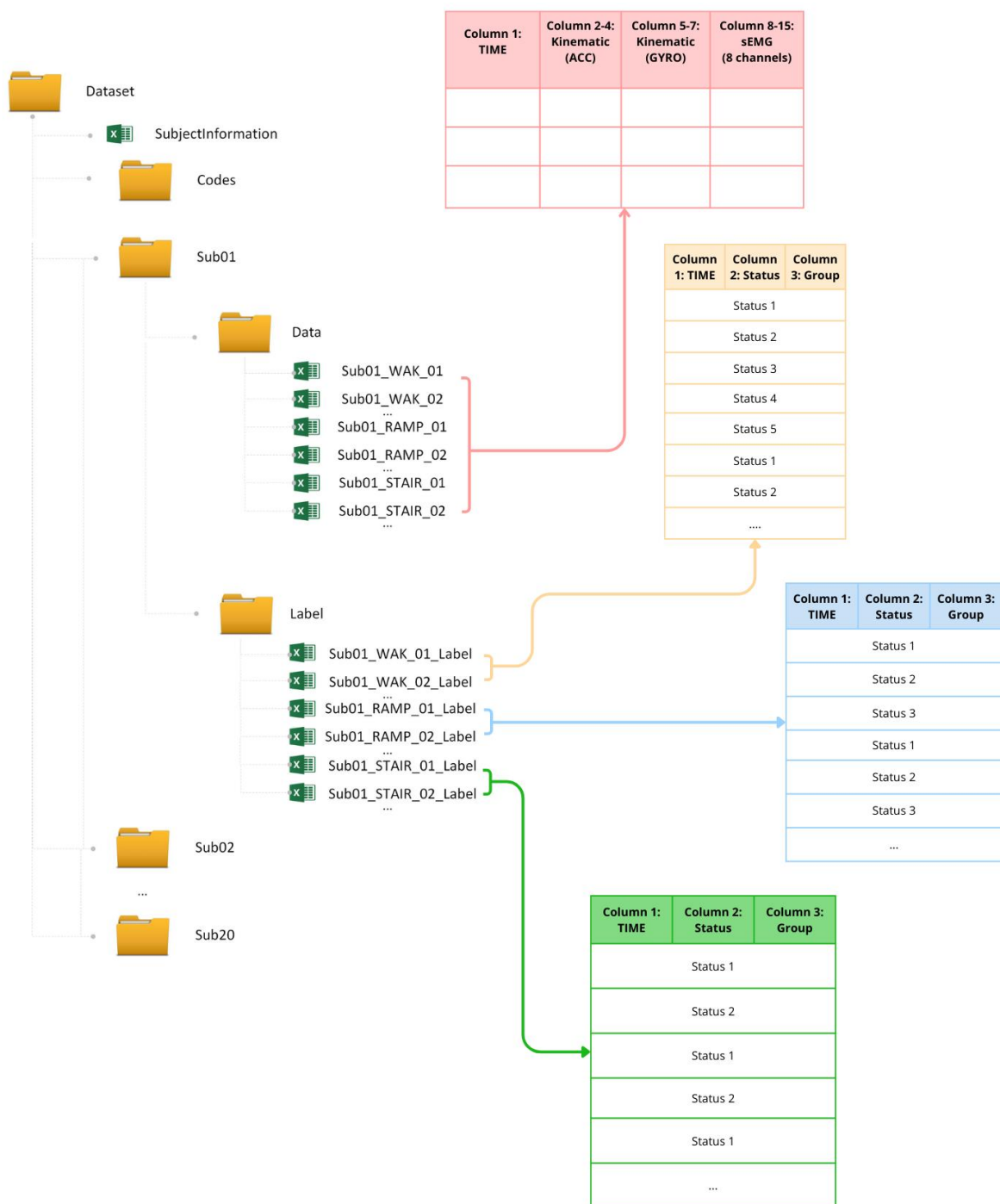


Figura 19. Rappresentazione grafica del dataset completo.

## 3.2 Segmentazione delle fasi del ciclo del passo

Le varie fasi inerenti al ciclo del passo sono state trovate a partire dai segnali dei soggetti acquisiti nelle varie prove, utilizzando i protocolli descritti nei capitoli precedenti.

In particolare, per le prove eseguite in piano, si sono individuati i punti di *initial contact* (IC), *heel rise* (HR), *toe off* (TO), *feet adjacent* (FA) e *tibia vertical* (TBV) per tutta la durata della prova, come descritto in **Figura 20**. Per ogni prova relativa a ciascun soggetto, non si sono riscontrati particolari problemi nell'individuazione dei punti nel segnale in quanto, l'algoritmo che ha permesso la segmentazione, è basato su delle soglie che permettono correttamente di individuare la successione degli eventi che compongono il cammino a partire dalla velocità angolare fornita dal giroscopio.

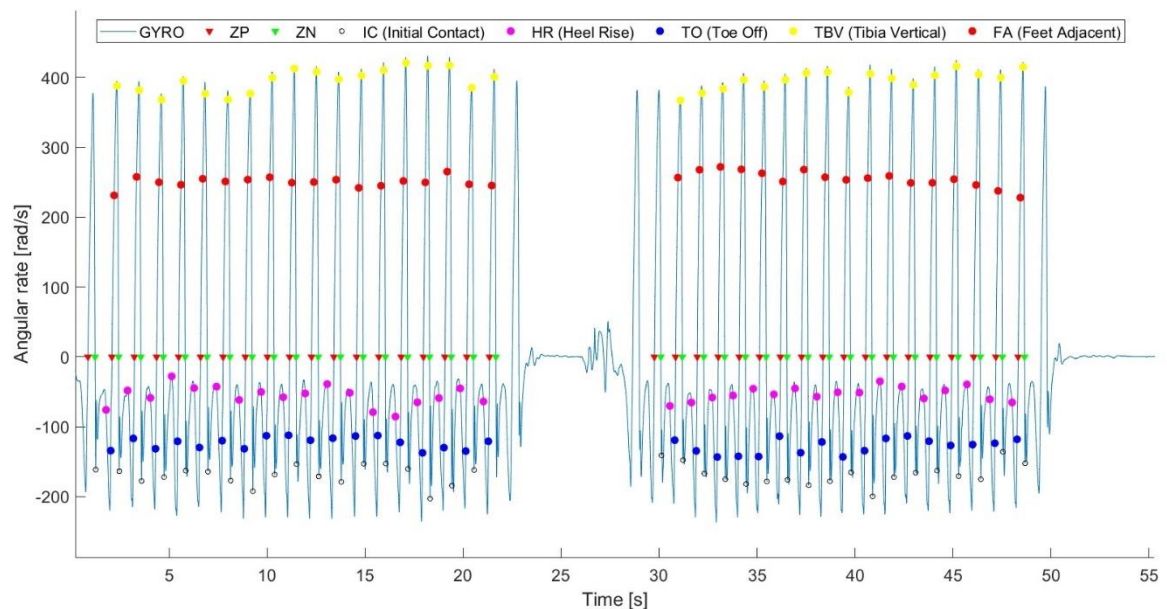
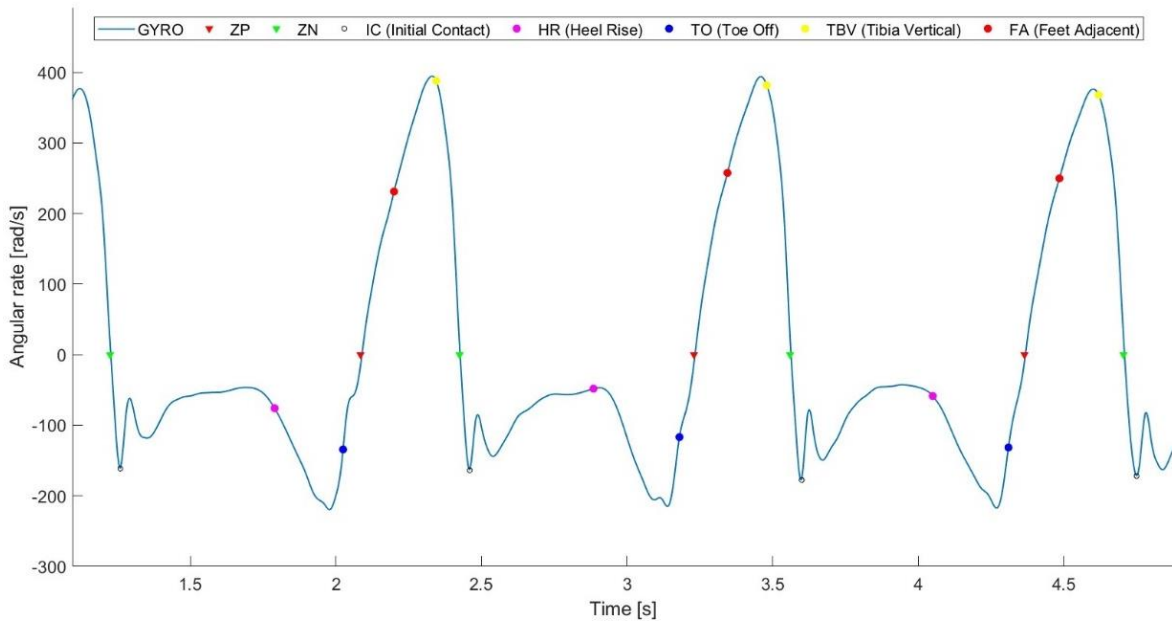


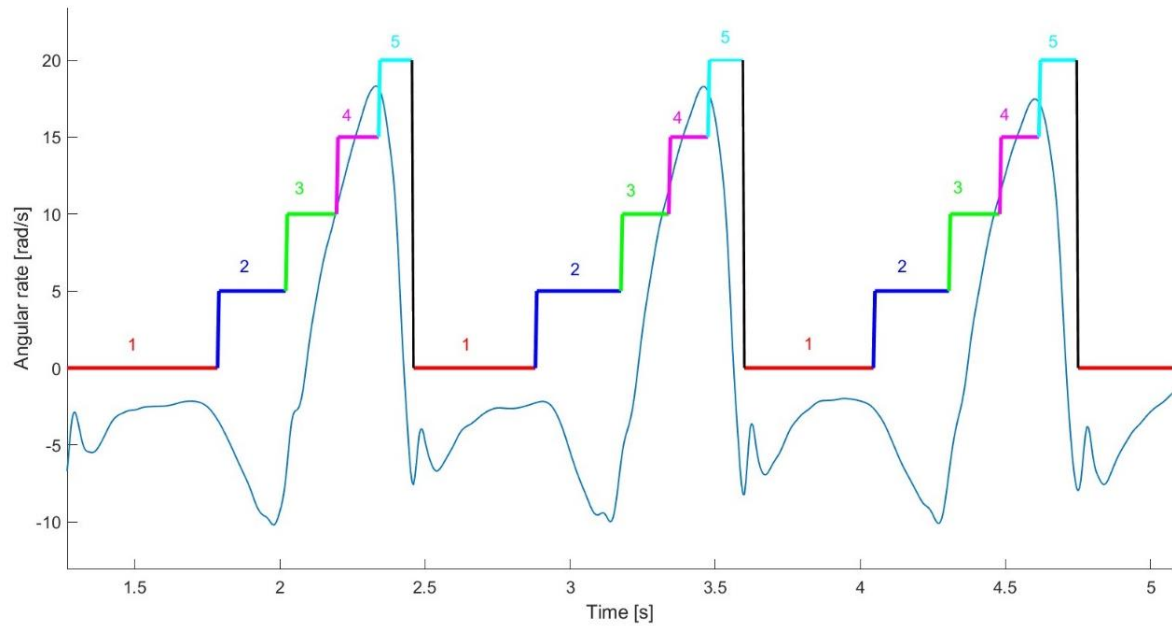
Figura 20. Esempio di individuazione delle fasi del ciclo del passo per le prove eseguite in piano.

La **Figura 21** mostra un dettaglio dei primi tre cicli del passo con i relativi punti individuati dall’algoritmo.



*Figura 21. Dettaglio di individuazione delle fasi nei primi cicli del passo per le prove eseguite in piano.*

In **Figura 22** vengono visualizzate graficamente le label individuate e sovrapposte al segnale normalizzato. Questa modalità di sovrapposizione “a gradino” permette di associare con facilità le varie fasi della camminata per tutte le prove eseguite dal soggetto e a capire la durata temporale di ciascuna di esse.



*Figura 22. Esempio di individuazione delle label sovrapposte al segnale per le prove eseguite in piano.*

Per le prove eseguite su superficie inclinata, si sono individuati, invece, i punti di *heel strike* (HS), *toe off* (TO) e *mid-swing* (MS). La **Figura 23** mostra graficamente il segnale acquisito dal giroscopio su cui vengono sovrapposti i punti trovati. In **Figura 24**, invece, viene rappresentato un dettaglio del segnale relativo ai primi cicli del passo condotti dal soggetto con le relative fasi sovrapposte.

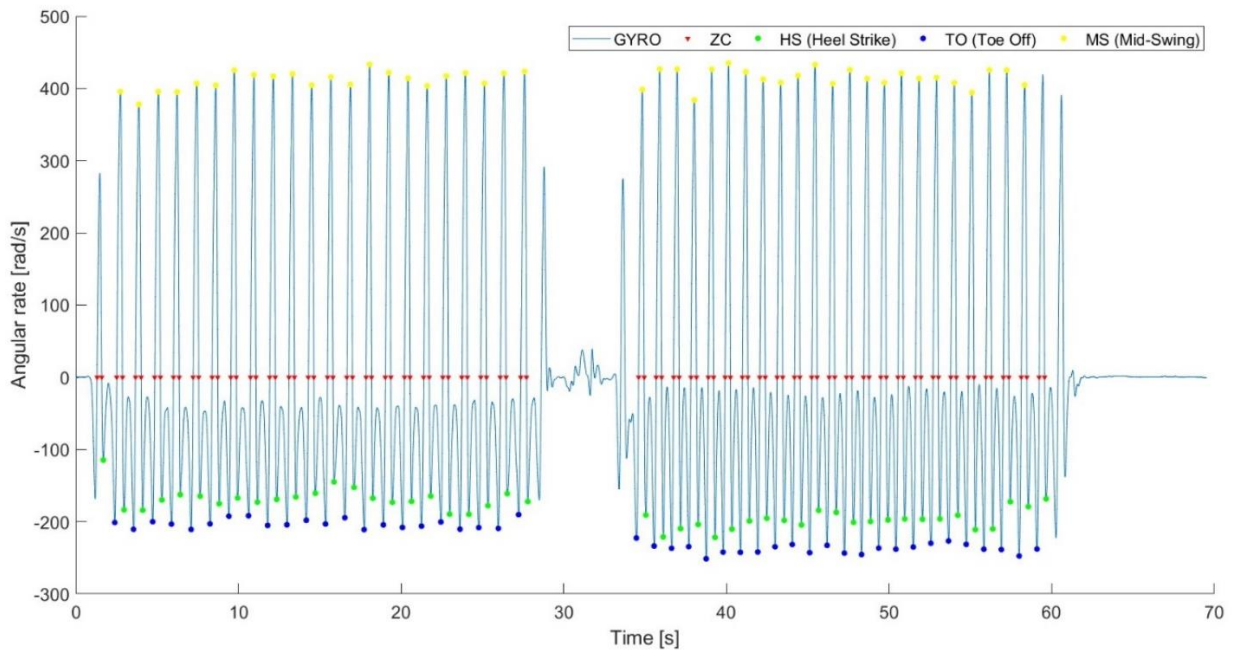


Figura 23. Esempio di individuazione delle fasi del ciclo del passo per le prove eseguite su piano inclinato.

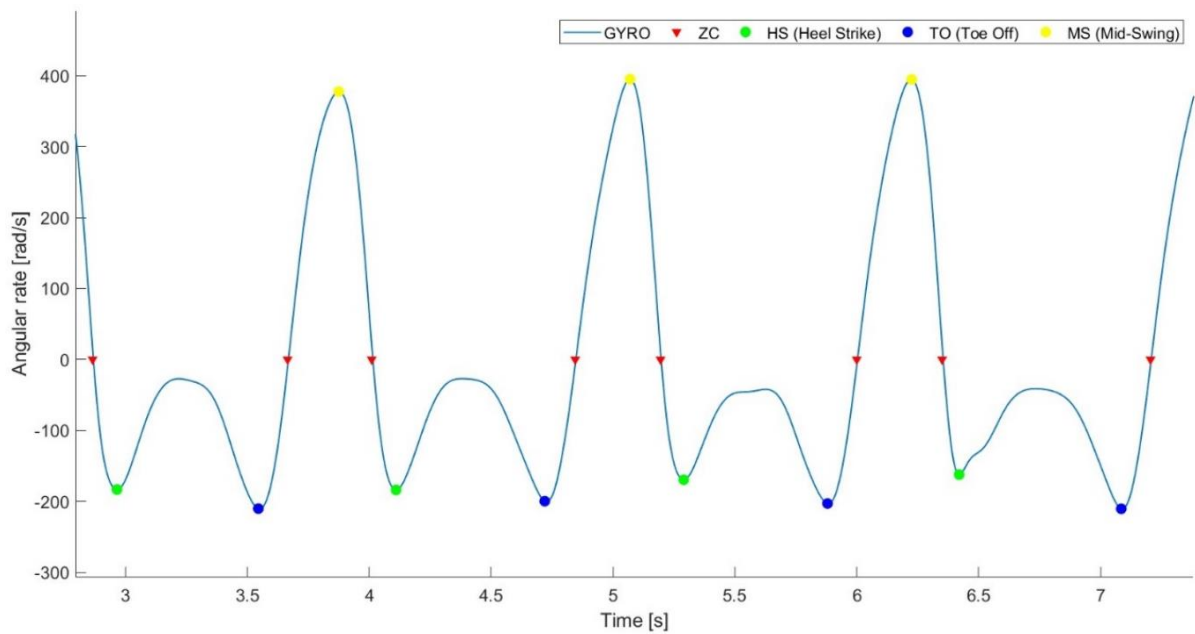
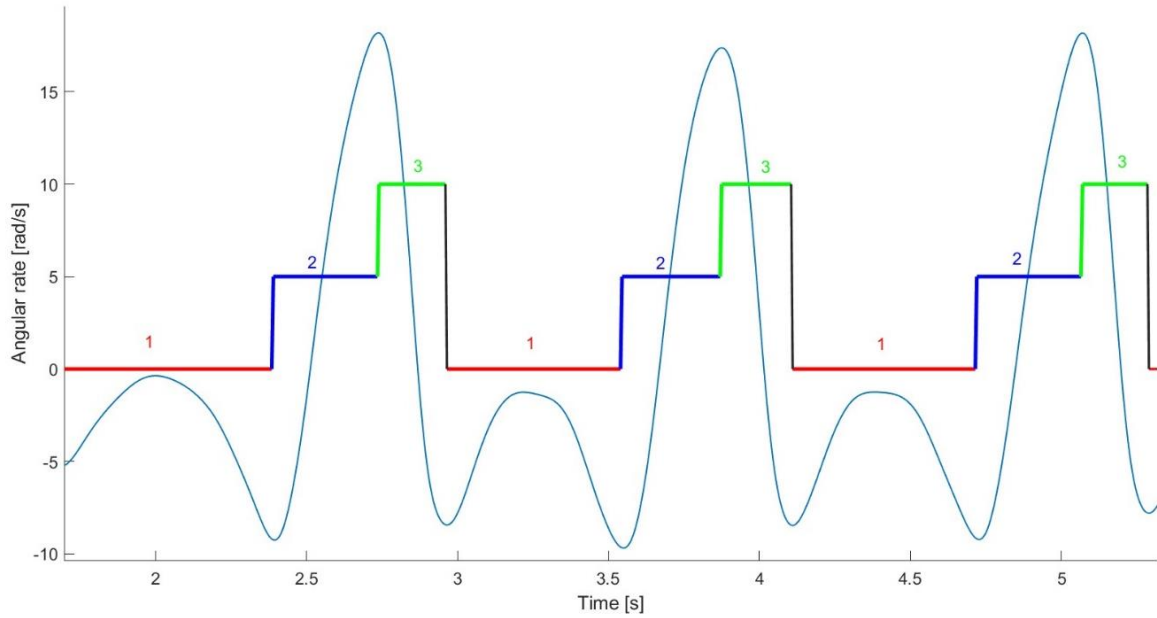


Figura 24. Dettaglio di individuazione delle fasi nei primi cicli del passo per le prove eseguite su piano inclinato.

Le etichette individuate che mostrano temporalmente il cambiamento di ogni fase sono riportate in **Figura 25**.



*Figura 25. Esempio di individuazione delle label sovrapposte al segnale per le prove eseguite su piano inclinato.*

Infine, per quanto riguarda le prove di salita e discesa di scale sono state individuate le fasi di *initial contact* (IC) e di *foot off* (FO), come mostrato in **Figura 26**.

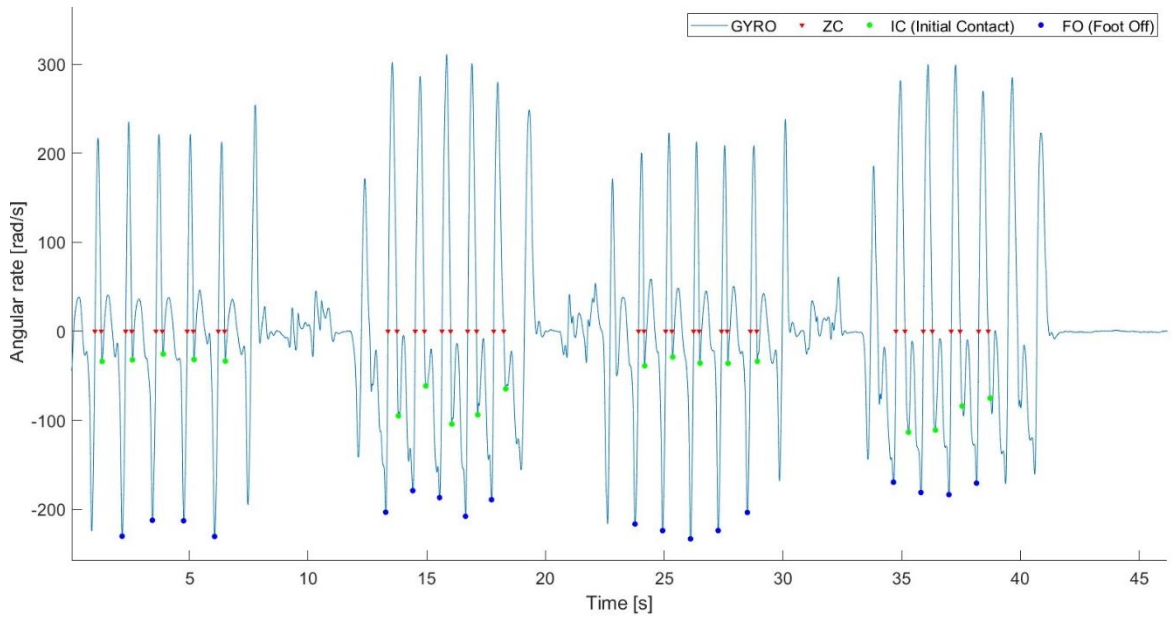


Figura 26. Esempio di individuazione delle fasi del ciclo del passo per le prove eseguite su scale.

In dettaglio si possono osservare i punti trovati relativi ai primi passi acquisiti sia rispetto alla prova eseguita in salita (**Figura 27**) che in discesa (**Figura 28**).

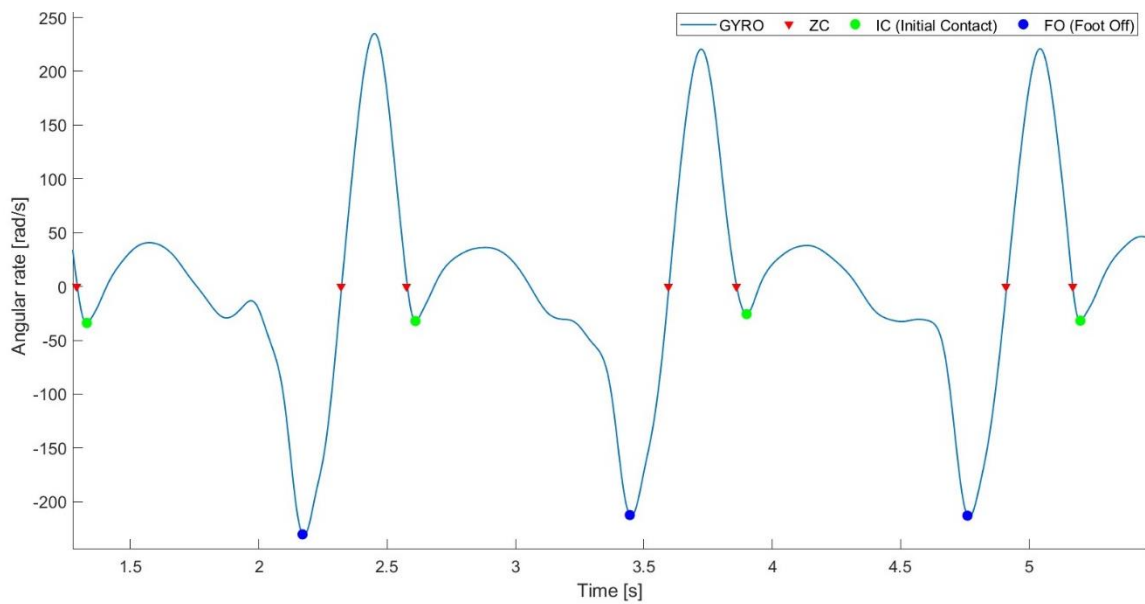


Figura 27. Dettaglio di individuazione delle fasi nei primi cicli del passo per le prove di salita di scale.

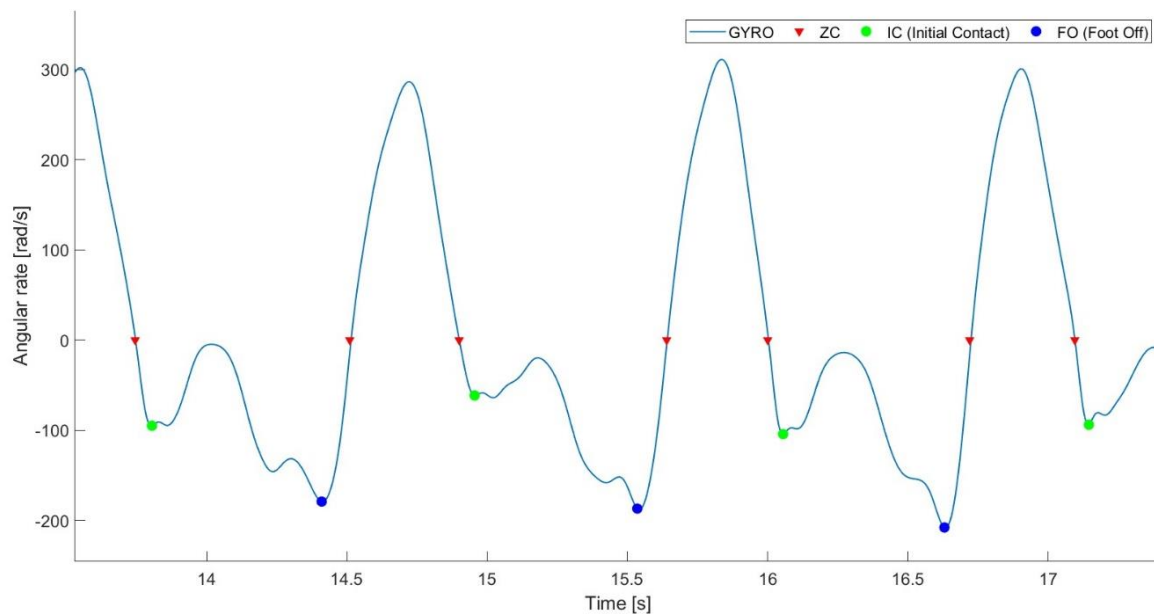


Figura 28. Dettaglio di individuazione delle fasi nei primi cicli del passo per le prove di discesa di scale.

Anche rispetto all'ultima modalit  di cammino, condotta su scale, si riporta la rappresentazione delle label sovrapposte al segnale del giroscopio (**Figura 29**).

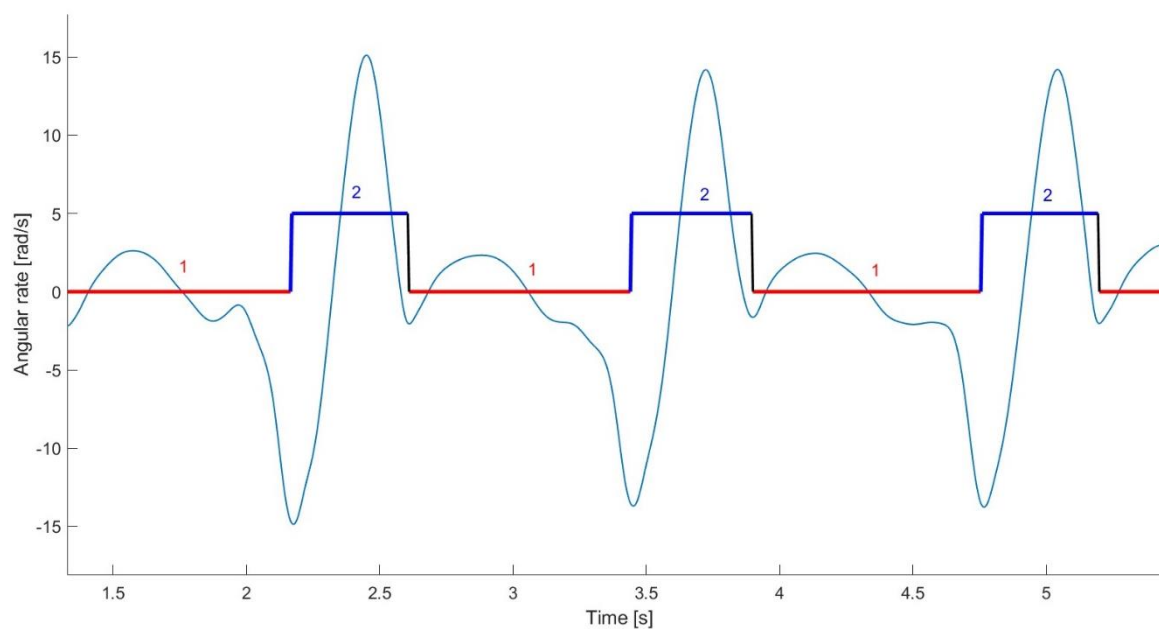


Figura 29. Esempio di individuazione delle label sovrapposte al segnale per le prove eseguite su scale.



# Capitolo 4

## Discussioni

Lo studio proposto ha riguardato la creazione di un dataset che contiene tutte le informazioni utili per l'implemento di modelli e algoritmi che possono essere utilizzati in ambito protesico al fine di migliorare le caratteristiche e le funzionalità di apparecchiature che possano risultare utili all'uomo nella quotidianità. A tal fine, sono stati considerati le informazioni fornite dal sensore inerziale, in particolar modo dal giroscopio, per segmentare il cammino di ogni soggetto nelle diverse fasi che caratterizzano il ciclo del passo. Per far ciò, sono risultati utili gli studi presi dalla letteratura che hanno rappresentato il punto di partenza per lo sviluppo di algoritmi efficaci ad individuare in ciascun soggetto e per ciascuna prova, tutti gli eventi desiderati. Le soglie stabilite negli algoritmi, sia in termini temporali che in ampiezza, si sono rivelate efficaci nell'individuazione dei punti di interesse, anche in condizioni variabili del cammino. Questo risultato è stato ottenuto anche in situazioni caratterizzate da un'elevata presenza di rumore sovrapposto al segnale acquisito, come nel caso della salita e discesa di scale. La scelta di non considerare i cicli del passo immediatamente precedenti e successivi ai cambi di direzione, così come l'esclusione del primo e dell'ultimo passo, rende l'individuazione delle fasi ancora più precisa. Questo approccio evita di considerare parti del segnale in cui il soggetto si trova nella fase iniziale o conclusiva del cammino, momenti in cui il rumore sovrapposto al segnale è

particolarmente elevato e l'identificazione dei punti potrebbe risultare imprecisa o eccessivamente variabile.

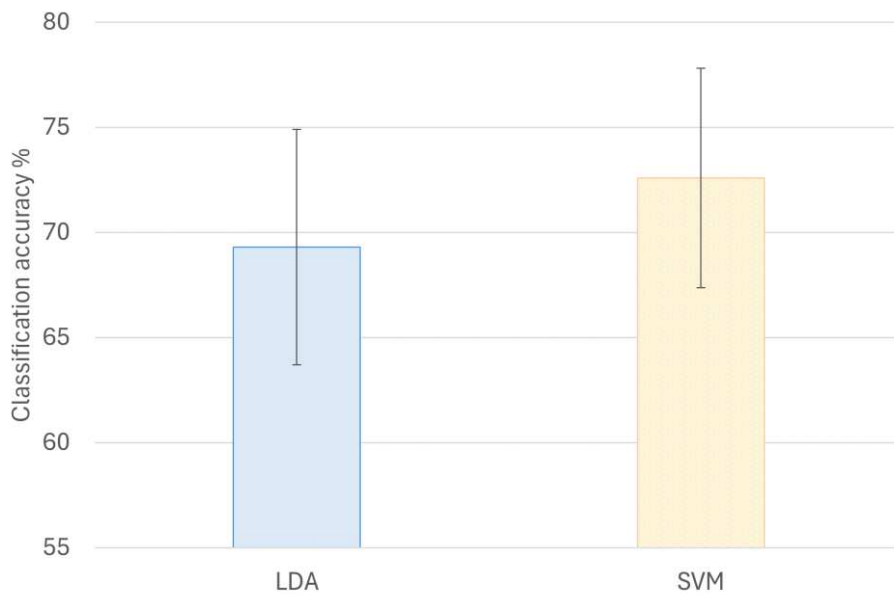
Il metodo CSAV, proposto da Salminen et al. (2024) [28], è stato efficace nell'individuazione di cinque fasi nelle prove eseguite in piano, grazie alle percentuali relative all'area cumulativa del segnale fornite sperimentalmente e ai metodi di zero-crossing, che hanno consentito di trovare con precisione tutti i punti d'interesse. La particolarità di questo metodo è il fatto di aver permesso l'individuazione dei punti HR, che solo in pochi studi è stata presa in considerazione. È importante sottolineare il ruolo cruciale dell'individuazione del punto IC, che in questo algoritmo, come in molti altri, costituisce il punto di partenza per la ricerca degli altri eventi chiave che caratterizzano il cammino.

Rispetto al metodo proposto da Gouwanda e Gopalai (2015) [23], che ha permesso di individuare tre fasi relative al ciclo del passo nelle prove eseguite su superficie inclinata, l'algoritmo sviluppato è risultato utile per individuare dei punti chiave per l'analisi del cammino. In questo caso si sono utilizzati dei metodi euristici uniti a degli algoritmi di zero-crossing, considerando i punti di minimo e di massimo del segnale e associandoli alle diverse fasi d'interesse. La difficoltà nell'impiego di questo tipo di metodo risiede nel fatto che gli algoritmi basati sull'individuazione dei punti di massimo e minimo del segnale possono commettere errori a causa dell'elevata variabilità del segnale tra un ciclo del passo e l'altro, sia all'interno dello stesso soggetto che tra soggetti diversi. Questa problematica è stata risolta considerando delle soglie temporali e applicando dei filtri al segnale in modo tale da ridurre il rumore sovrapposto.

L'ultimo algoritmo sviluppato per le prove relative alla salita e alla discesa di scale fa riferimento all'articolo proposto da Catalfamo et al. (2014) [56]. È stato possibile

individuare i punti IC e FO grazie a metodi di zero-crossing e d'individuazione di minimi e massimi. I punti di contatto iniziale e di sollevamento del piede risultano infatti essere eventi chiave per quanto riguarda l'analisi del cammino. Anche in questo caso, c'è stato il bisogno di filtrare il segnale e di utilizzare delle soglie temporali che hanno permesso di minimizzare l'errore rispetto all'individuazione dei punti.

I tre algoritmi proposti si sono dimostrati efficaci nella creazione di etichette grazie alle quali è possibile identificare le diverse fasi e la loro sequenza nel tempo, per tutti i soggetti e le prove effettuate. La creazione di etichette rappresenta un'innovazione rispetto alla maggior parte dei dataset presenti in letteratura, che spesso non le riportano. Esse risultano invece fondamentali per condurre processi di classificazione e analisi del cammino umano in modo accurato ed efficiente. Le etichette sono quindi state aggiunte al dataset in una cartella apposita per ogni soggetto, in modo tale che il dataset finale contenesse informazioni sia riguardanti i dati acquisiti dal Myo Armband, sia rispetto alle fasi trovate per segmentare il ciclo del passo. I dati forniti dall'IMU e l'individuazione delle fasi del ciclo del passo attraverso i segnali forniti dal giroscopio sono risultati, inoltre, utili al fine di sincronizzare e segmentare il segnale EMG. Questo consente di associare specifiche porzioni del segnale elettromiografico a fasi specifiche del ciclo del passo durante la locomozione dei soggetti. L'estrazione delle caratteristiche è risultata importante al fine di avere un insieme di informazioni utilizzabili per la classificazione delle fasi. In **Figura 30** è riportato il grafico dei risultati di classificazione utilizzando l'algoritmo SVM e LDA delle fasi del cammino eseguito in piano mediante i segnali EMG acquisiti dal Myo Armband. Sono riportati i risultati della classificazione di tutte le prove di camminata in piano di tutti i soggetti riportando la media e la deviazione standard.



*Figura 30. Accuratezza della classificazione delle fasi del ciclo del passo utilizzando il segnale EMG.*

Dal grafico di classificazione riportato, prendendo in considerazione il setup sperimentale che prevede l'utilizzo di un solo sensore che lavora a basse frequenze e considerando la media e la deviazione standard rispetto a tutti i 20 soggetti, si può affermare che i risultati ottenuti sono soddisfacenti. Sarebbe opportuno sviluppare degli algoritmi che migliorino i risultati di classificazione e sviluppare dispositivi facilmente indossabili che lavorino anche a più alte frequenze. Come dimostrato in letteratura, l'uso di frequenze più alte, come nel caso di dispositivi elettromiografici con una frequenza di campionamento di 1000 Hz, consentirebbe drastici miglioramenti nelle prestazioni di classificazione [49].

Nonostante le considerazioni fatte rispetto alla segmentazione delle fasi eseguita tramite l'IMU e ai risultati di classificazione del segnale EMG, i dati raccolti e l'elaborazione di questi, sono stati utili al fine di sviluppare un dataset organizzato e facilmente accessibile.

Questi tipi di studio offrono l'accesso a dei dati e a delle informazioni legate al cammino dell'uomo che risultano cruciali in ricerca per lo sviluppo di nuove tecnologie che possono essere fondamentali nella vita di molte persone.

## Conclusioni

In questo elaborato sono state descritte le procedure di un protocollo sperimentale, includendo l'analisi e l'elaborazione dei dati acquisiti attraverso il dispositivo Myo Armband, che ha come obiettivo la creazione di un dataset che contenga informazioni per lo sviluppo e l'implementazione di algoritmi utili in ambito protesico. I dati relativi all'IMU e ai sensori EMG sono stati raccolti da 20 soggetti che hanno indossato il bracciale sensorizzato sulla gamba dominante in tre diverse modalità di cammino (in piano, su superficie inclinata e discesa/salita di scale). L'integrazione di informazioni sia inerziali che elettromiografiche potrebbe migliorare in modo significativo l'efficacia e la precisione dei sistemi di controllo protesico, offrendo soluzioni più naturali e funzionali per gli utenti.

Il dataset è stato sviluppato con l'obiettivo di essere accessibile a tutti e di includere i dati necessari per consentire analisi approfondite che riguardano la deambulazione umana. Includendo, oltre ai dati forniti dal sensore inerziale, anche le informazioni elettromiografiche, il dataset può favorire lo sviluppo di protesi mioelettriche controllate dal

sistema nervoso. Questa tipologia di protesi, già ampiamente sviluppata per gli arti superiori, rappresenta ancora un'innovazione per gli arti inferiori, offrendo però la possibilità di compiere un maggior numero di movimenti e di possedere più ampie funzionalità.

In prospettiva futura, dataset come quello presentato, dovrebbero considerare un maggior numero di partecipanti, che siano rappresentativi della variabilità della popolazione. Inoltre, si dovrebbe considerare un maggior numero di movimenti che possano essere significativi nel monitoraggio dei segnali in tutte le condizioni presenti nella quotidianità. La ricerca in questo ambito, oltre a raccogliere un numero maggiore di dataset disponibili per analisi future, dovrebbe promuovere lo sviluppo di algoritmi e modelli avanzati, stabilire protocolli di misura precisi e sviluppare sensori indossabili integrati e di piccole dimensioni, utili per il controllo delle protesi degli arti inferiori.

Fino ad ora l'analisi del cammino e del movimento dell'uomo ha permesso di raggiungere grandi risultati, permettendo di migliorare lo stile di vita di molte persone. La ricerca e lo studio in questo settore, attraverso la creazione di dataset come quello presentato e l'implementazione di algoritmi di analisi sempre più innovativi, possono favorire lo sviluppo di tecnologie avanzate, in grado di produrre protesi di alta qualità che consentano di sostituire la normale deambulazione in tutti i contesti quotidiani.



# Appendice - Codici Matlab

## Codice Matlab SaveData.m

```
% SaveData.m saves and organizes the acquired data into a table containing
% accelerometer, gyroscope, and EMG data, and then saves the entire table to an Excel file.

maxTrials = 8;

% Change with the path of the folder where the data is taken from
baseFolder = 'C:\Users\39346\Desktop\Dataset';
% Change with the destination folder path
destFolder = 'C:\Users\39346\Desktop\DatasetFile';

% Level ground walking
for subject = 1:20
    subjectField = sprintf('Sub%02d', subject);
    subjectDestFolder = fullfile(destFolder, sprintf('Sub%02d', subject), 'Data');
    if ~exist(subjectDestFolder, 'dir')
        mkdir(subjectDestFolder);
    end
    for trial = 1:maxTrials
        fileName = sprintf('%s/Sub%02d/Data/Sub%02d_WAK_%02d.mat', baseFolder, subject, subject, trial);
        if exist(fileName, 'file')
            data = load(fileName);

            % Resample
            ACC_data_resampled = [interp(data.ACC_data(:, 1), 4), interp(data.ACC_data(:, 2), 4), interp(data.ACC_data(:, 3), 4)];
            GYRO_data_resampled = [interp(data.GYRO_data(:, 1), 4), interp(data.GYRO_data(:, 2), 4), interp(data.GYRO_data(:, 3), 4)];

            % Synchronization of data lengths
            min_length = min([length(ACC_data_resampled), length(GYRO_data_resampled), length(data.EMG_data)]);
            ACC_data_resampled = ACC_data_resampled(1:min_length, :);
            GYRO_data_resampled = GYRO_data_resampled(1:min_length, :);
            EMG_data = data.EMG_data(1:min_length, :);
        end
    end
end
```

```

        TIME_data = data.TIME_data(1:min_length, :);
        TIME_data=(0:0.005:(size(EMG_data(:,1),1)-1)*0.005)'; %↵
rebuilding the correct time vector

        t = table(TIME_data, ...
                ACC_data_resampled(:,1), ACC_data_resampled(:,2), ↵
ACC_data_resampled(:,3), ...
                GYRO_data_resampled(:,1), GYRO_data_resampled(:, ↵
2), GYRO_data_resampled(:,3), ...
                EMG_data(:,1), EMG_data(:,2), EMG_data(:,3), ↵
EMG_data(:,4), ...
                EMG_data(:,5), EMG_data(:,6), EMG_data(:,7), ↵
EMG_data(:,8));

        t.Properties.VariableNames = {'Time', 'Kinematic:ACC_x', ↵
'Kinematic:ACC_y', 'Kinematic:ACC_z', 'Kinematic:GYRO_x', 'Kinematic:↵
GYRO_y', 'Kinematic:GYRO_z', 'sEMG:electrode1', 'sEMG:electrode2', ↵
'sEMG:electrode3', 'sEMG:electrode4', 'sEMG:electrode5', 'sEMG:↵
electrode6', 'sEMG:electrode7', 'sEMG:electrode8'};

        outputFile = fullfile(subjectDestFolder, sprintf('Sub%↵
02d_WAK_%02d.xls', subject, trial));
        writetable(t, outputFile, 'FileType', 'spreadsheet');
        else
            break;
        end
    end
end
end

% Ramp
for subject = 1:20
    subjectField = sprintf('Sub%02d', subject);
    subjectDestFolder = fullfile(destFolder, sprintf('Sub%02d', ↵
subject), 'Data');
    if ~exist(subjectDestFolder, 'dir')
        mkdir(subjectDestFolder);
    end
    for trial = 1:maxTrials

```

```

        fileName = sprintf('%s/Sub%02d/Data/Sub%02d_RAMP_%02d.mat',
baseFolder, subject, subject, trial);
        if exist(fileName, 'file')
            data = load(fileName);

            % Resample
            ACC_data_resampled = [interp(data.ACC_data(:, 1), 4), interp
(data.ACC_data(:, 2), 4), interp(data.ACC_data(:, 3), 4)];
            GYRO_data_resampled = [interp(data.GYRO_data(:, 1), 4),
interp(data.GYRO_data(:, 2), 4), interp(data.GYRO_data(:, 3), 4)];

            % Synchronization of data lengths
            min_length = min([length(ACC_data_resampled), length
(GYRO_data_resampled), length(data.EMG_data)]);
            ACC_data_resampled = ACC_data_resampled(1:min_length, :);
            GYRO_data_resampled = GYRO_data_resampled(1:min_length, :);
            EMG_data = data.EMG_data(1:min_length, :);
            TIME_data = data.TIME_data(1:min_length, :);
            TIME_data=(0:0.005:(size(EMG_data(:,1),1)-1)*0.005)'; %
rebuilding the correct time vector

            t = table(TIME_data, ...
                ACC_data_resampled(:,1), ACC_data_resampled(:,2),
ACC_data_resampled(:,3), ...
                GYRO_data_resampled(:,1), GYRO_data_resampled(:,
2), GYRO_data_resampled(:,3), ...
                EMG_data(:,1), EMG_data(:,2), EMG_data(:,3),
EMG_data(:,4), ...
                EMG_data(:,5), EMG_data(:,6), EMG_data(:,7),
EMG_data(:,8));

            t.Properties.VariableNames = {'Time', 'Kinematic:ACC_x',
'Kinematic:ACC_y', 'Kinematic:ACC_z', 'Kinematic:GYRO_x', 'Kinematic:
GYRO_y', 'Kinematic:GYRO_z', 'sEMG:electrode1', 'sEMG:electrode2',
'sEMG:electrode3', 'sEMG:electrode4', 'sEMG:electrode5', 'sEMG:
electrode6', 'sEMG:electrode7', 'sEMG:electrode8'};
            outputFile = fullfile(subjectDestFolder, sprintf('Sub%
02d_RAMP_%02d.xls', subject, trial));
            writetable(t, outputFile, 'FileType', 'spreadsheet');
        else

            break;
        end
    end
end
end

```

```

% Stairs
for subject = 1:20
    subjectField = sprintf('Sub%02d', subject);
    subjectDestFolder = fullfile(destFolder, sprintf('Sub%02d',
subject), 'Data');
    if ~exist(subjectDestFolder, 'dir')
        mkdir(subjectDestFolder);
    end

    for trial = 1:maxTrials
        fileName = sprintf('%s/Sub%02d/Data/Sub%02d_STAIR_%02d.mat',
baseFolder, subject, subject, trial);
        if exist(fileName, 'file')
            data = load(fileName);

            % Resample
            ACC_data_resampled = [interp(data.ACC_data(:, 1), 4), interp
(data.ACC_data(:, 2), 4), interp(data.ACC_data(:, 3), 4)];
            GYRO_data_resampled = [interp(data.GYRO_data(:, 1), 4),
interp(data.GYRO_data(:, 2), 4), interp(data.GYRO_data(:, 3), 4)];

            % Synchronization of data lengths
            min_length = min([length(ACC_data_resampled), length
(GYRO_data_resampled), length(data.EMG_data)]);
            ACC_data_resampled = ACC_data_resampled(1:min_length, :);
            GYRO_data_resampled = GYRO_data_resampled(1:min_length, :);
            EMG_data = data.EMG_data(1:min_length, :);
            TIME_data = data.TIME_data(1:min_length, :);
            TIME_data=(0:0.005:(size(EMG_data(:,1),1)-1)*0.005)'; %
rebuilding the correct time vector

            t = table(TIME_data, ...

                ACC_data_resampled(:,1), ACC_data_resampled(:,2),
ACC_data_resampled(:,3), ...
                GYRO_data_resampled(:,1), GYRO_data_resampled(:,
2), GYRO_data_resampled(:,3), ...
                EMG_data(:,1), EMG_data(:,2), EMG_data(:,3),
EMG_data(:,4), ...
                EMG_data(:,5), EMG_data(:,6), EMG_data(:,7),
EMG_data(:,8));

            t.Properties.VariableNames = {'Time', 'Kinematic:ACC_x',
'Kinematic:ACC_y', 'Kinematic:ACC_z', 'Kinematic:GYRO_x', 'Kinematic:
GYRO_y', 'Kinematic:GYRO_z', 'sEMG:electrode1', 'sEMG:electrode2',
'sEMG:electrode3', 'sEMG:electrode4', 'sEMG:electrode5', 'sEMG:
electrode6', 'sEMG:electrode7', 'sEMG:electrode8'};
            outputFile = fullfile(subjectDestFolder, sprintf('Sub%
02d_STAIR_%02d.xls', subject, trial));
            writetable(t, outputFile, 'FileType', 'spreadsheet');
        else

            break;
        end
    end
end
end

```

## Codice Matlab Segmentation\_WAK.m

```
% This code processes gyroscope data (GYRO_data_y) and identifies various
% gait cycle events:
% Initial Contact (IC), Heel Rise (HR), Toe Off (TO), Feet Adjacent (FA) and Tibia Vertical (TBV).
% A vector called label is created where the first column contains the time,
% the second column assigns 1 for the phase between IC and HR, 2 for HR to TO, 3 for TO to FA,
% 4 for FA to TBV, and 5 for TBV to the next IC.
% The third column always contains 1, indicating the group of movements according to the methodology used.
% For each subject it is possible to visualize each trial.

subjectID = input('Enter Subject ID (01 to 20): ', 's');
trialNum = input('Enter the trial number (e.g. 01): ', 's');

% change with the your Dataset folder (RAw Data)
load(sprintf('%s\\Sub%s\\Data\\Sub%s_WAK_%s.mat', 'C:\Users\39346\Desktop\Dataset', subjectID, subjectID, trialNum));

if length(ACC_data) ~= length(GYRO_data) || length(ACC_data) ~= length(EMG_data) || length(GYRO_data) ~= length(EMG_data)
% Resample
ACC_data = [interp(ACC_data(:, 1), 4), interp(ACC_data(:, 2), 4), interp(ACC_data(:, 3), 4)];
GYRO_data= [interp(GYRO_data(:, 1), 4), interp(GYRO_data(:, 2), 4), interp(GYRO_data(:, 3), 4)];
% Synchronization of data lengths
min_length = min([length(ACC_data), length(GYRO_data), length(EMG_data)]);
ACC_data = ACC_data(1:min_length, :);
GYRO_data = GYRO_data(1:min_length, :);
EMG_data = EMG_data(1:min_length, :);
TIME_data = TIME_data(1:min_length, :);
end
TIME_data=(0:0.005:(size(EMG_data(:,1),1)-1)*0.005)'; %rebuilding the

correct time vector
GYRO_data_y= -GYRO_data(:,2);
SR=200;
%Correction_factors
HR_Correction_Factor = @(stride_time) 0.156 - 0.154 * stride_time;
FA_Correction_Factor = @(ZPcp) -0.254 + 0.384 * ZPcp;
% zero-crossings
ZP = findZeroCrossings(GYRO_data_y, 'positive');
ZN = findZeroCrossings(GYRO_data_y, 'negative');
```

```

% Initialize valid zero-crossings
ZP_valid = [];
ZN_valid = [];
% Threshold for signal segmentation
th=max(GYRO_data_y)/2;
for i=1:length(ZN)
    if ZN(1) < ZP(1) && i == 1
        % If the first point found is a ZN, skip the first iteration
        continue;
    elseif ZN(1) < ZP(1)
        % If ZN appears before ZP, use GYRO_data_y(ZP(i-1):ZN(i))
        segnale = GYRO_data_y(ZP(i-1):ZN(i));
    else
        % Otherwise, use GYRO_data_y(ZP(i):ZN(i))
        segnale = GYRO_data_y(ZP(i):ZN(i));
    end

    % Check if maximum value of segment exceeds the threshold
    if max(segnale) >= th
        if ZN(1) < ZP(1) && i > 1
            ZP_valid = [ZP_valid; ZP(i-1)];
            ZN_valid = [ZN_valid; ZN(i)];
        else
            ZP_valid = [ZP_valid; ZP(i)];
            ZN_valid = [ZN_valid; ZN(i)];
        end
    end
end

% Remove ZN and ZP if the interval between them is greater than 2✓
seconds
i = 1;

while i < length(ZN_valid)
    if TIME_data(ZP_valid(i+1)) - TIME_data(ZN_valid(i)) > 2
        % Remove ZN and ZP
        ZN_valid(i:i+1) = [];
        ZP_valid(i:i+1) = [];

        i = i + 1;
    else
        i = i + 1;
    end
end

% Remove the last ZP and ZN
ZP_valid(end) = [];
ZN_valid(end) = [];

```

```

% derivate of GYRO_data_y
GYRO_data_y_diff = diff(GYRO_data_y); %points where the signal changes
%first minimum after each ZN_valid point
% Identify initial contact points (IC)
IC = [];
for i = 1:length(ZN_valid)
    zn = ZN_valid(i);
    for j = zn:length(GYRO_data_y_diff)-1
        if GYRO_data_y_diff(j) < 0 && GYRO_data_y_diff(j + 1) >= 0
            IC = [IC; j+1];
            break;
        end
    end
end
int = TIME_data(IC(2:end)) - TIME_data(IC(1:end-1));

% cumulative area for each interval between ZN_valid and the next ✓
ZP_valid
CSAVTS = [];
HR_points = [];
HR_times = [];
TO_points = [];
TO_times = [];

for i = 1:length(ZN_valid)-1

    zp = ZP_valid(i+1);
    zpcp=(zp-IC(i))/(IC(i+1)-IC(i));
    CF=round(FA_Correction_Factor(zpcp)*SR);
    if (TIME_data(zn) - TIME_data(zp)) >= 3
        continue;
    end
    interval_signal = GYRO_data_y(zp:zn);
    interval_time = TIME_data(zp:zn);
    cumulative_area = cumtrapz(interval_time, abs(interval_signal));
    total_area = cumulative_area(end);
    % cumulative area for swing phase
    CSAVSW = [CSAVSW; total_area];
    target_area_FA = 0.20 * total_area;
    target_area_TBV = 0.731 * total_area;
    FA_index = find(cumulative_area >= target_area_FA & GYRO_data_y(zp:✓
zn) > 100, 1) + zp - 1 - CF;
    TBV_index = find(cumulative_area >= target_area_TBV & GYRO_data_y✓
(zp:zn) > 100, 1) + zp - 1;

    % Verify that the FA and TBV points are above 100 on the y-axis
    if ~isempty(FA_index) && ~isempty(TBV_index) && GYRO_data_y✓
(FA_index) > 100 && GYRO_data_y(TBV_index) > 100
        FA_points = [FA_points; GYRO_data_y(FA_index)];
        FA_times = [FA_times; TIME_data(FA_index)];
        TBV_points = [TBV_points; GYRO_data_y(TBV_index)];
        TBV_times = [TBV_times; TIME_data(TBV_index)];
    end
end
end

```

```

% Correction of IC points
IC_diff_times = diff(TIME_data(IC));
valid_IC_indices = true(length(IC), 1);
for i = 1:length(IC_diff_times)
    if IC_diff_times(i) > 5
        % If the time difference is more than 5 seconds, delete the
first IC
        valid_IC_indices(i) = false;
    end
end
valid_IC_indices(1) = true; % Keep the first IC point as valid
IC = IC(valid_IC_indices); % Update IC to keep only valid points
IC_times = TIME_data(IC); % corresponding times of IC points

    if ~isempty(start_index) && ~isempty(end_index) && TO_times(i) -
HR_times(i) <= 3
        label(start_index:end_index, 2) = 2;
    end
end
if i <= length(TO_times) && i <= length(FA_times)
    % TO-FA
    start_index = find(TIME_data >= TO_times(i), 1);
    end_index = find(TIME_data < FA_times(i), 1, 'last');
    if ~isempty(start_index) && ~isempty(end_index) && FA_times(i) -
TO_times(i) <= 3
        label(start_index:end_index, 2) = 3;
    end
end
if i <= length(FA_times) && i <= length(TBV_times)
    % FA-TBV
    start_index = find(TIME_data >= FA_times(i), 1);
    end_index = find(TIME_data < TBV_times(i), 1, 'last');
    if ~isempty(start_index) && ~isempty(end_index) && TBV_times(i) -
FA_times(i) <= 3
        label(start_index:end_index, 2) = 4;
    end
end
if i <= length(TBV_times) && i < length(IC)
    % TBV-IC
    start_index = find(TIME_data >= TBV_times(i), 1);
    end_index = find(TIME_data < TIME_data(IC(i + 1)), 1, 'last');
    if ~isempty(start_index) && ~isempty(end_index) && TIME_data(IC
(i + 1)) - TBV_times(i) <= 3
        label(start_index:end_index, 2) = 5;
    end
end
end

% Manage intervals that are too long
for i = 1:length(IC)
    if i < length(HR_times)
        if HR_times(i) - TIME_data(IC(i)) > 3
            next_start = TIME_data(IC(i + 1));
            if i + 1 <= length(HR_times)
                next_end = HR_times(i + 1);
            end
        end
    end
end

```

```

        label(TIME_data >= next_start & TIME_data < next_end, 2) ✓
= 1;
    end
end
end

if i < length(HR_times) && i < length(TO_times)
    if TO_times(i) - HR_times(i) > 3
        next_start = HR_times(i + 1);
        if i + 1 <= length(TO_times)
            next_end = TO_times(i + 1);
            label(TIME_data >= next_start & TIME_data < next_end, 2) ✓
= 2;
        end
    end
end

if i < length(TO_times) && i < length(FA_times)
    if FA_times(i) - TO_times(i) > 3

        next_start = TO_times(i + 1);
        if i + 1 <= length(FA_times)
            next_end = FA_times(i + 1);
            label(TIME_data >= next_start & TIME_data < next_end, 2) ✓
= 3;
        end
    end
end

if i < length(FA_times) && i < length(TBV_times)
    if TBV_times(i) - FA_times(i) > 3

        next_start = FA_times(i + 1);
        if i + 1 <= length(TBV_times)
            next_end = TBV_times(i + 1);
            label(TIME_data >= next_start & TIME_data < next_end, 2) ✓
= 4;
        end
    end
end

if i < length(TBV_times) && i < length(IC)
    if TIME_data(IC(i + 1)) - TBV_times(i) > 3

        next_start = TBV_times(i + 1);
        if i + 1 < length(IC)
            next_end = TIME_data(IC(i + 1));
            label(TIME_data >= next_start & TIME_data < next_end, 2) ✓
= 5;
        end
    end
end
end
end

```

```

max_GYRO = max(abs(GYRO_data_y));
GYRO_data_y_normalized = GYRO_data_y / max_GYRO * 20; % Normalizes the
data so that the maximum value is 20
HR_points_normalized = HR_points / max_GYRO * 20;
TO_points_normalized = TO_points / max_GYRO * 20;
TBV_points_normalized = TBV_points / max_GYRO * 20;
FA_points_normalized = FA_points / max_GYRO * 20;
figure;
plot(TIME_data, GYRO_data_y_normalized, 'DisplayName', 'GYRO_data_y');
hold on;
plot(TIME_data(ZP_valid), zeros(size(ZP_valid)), 'rv', 'MarkerSize', 5,
'MarkerFaceColor', 'r', 'DisplayName', 'ZP');
plot(TIME_data(ZN_valid), zeros(size(ZN_valid)), 'gv', 'MarkerSize', 5,
'MarkerFaceColor', 'g', 'DisplayName', 'ZN');
plot(IC_times, GYRO_data_y_normalized(IC), 'ko', 'MarkerSize', 4,
'DisplayName', 'IC (initial contact)');
plot(HR_times, HR_points_normalized, 'mo', 'MarkerSize', 6,
'MarkerFaceColor', 'm', 'DisplayName', 'HR (heel rise)');
plot(TO_times, TO_points_normalized, 'bo', 'MarkerSize', 6,
'MarkerFaceColor', 'b', 'DisplayName', 'TO (toe off)');
plot(TBV_times, TBV_points_normalized, 'yo', 'MarkerSize', 6,
'MarkerFaceColor', 'y', 'DisplayName', 'TBV (mid-swing)');
plot(FA_times, FA_points_normalized, 'ro', 'MarkerSize', 6,
'MarkerFaceColor', 'r', 'DisplayName', 'FA (initial swing)');

n = size(label, 1);
start_time_1 = [];
end_time_1 = [];
start_time_2 = [];

end_time_2 = [];
start_time_3 = [];
end_time_3 = [];
start_time_4 = [];
end_time_4 = [];
start_time_5 = [];
end_time_5 = [];
in_interval_1 = false;
in_interval_2 = false;
in_interval_3 = false;
in_interval_4 = false;
in_interval_5 = false;
last_y = NaN; % To keep track of the latest y value
last_end_time = NaN; % To track the latest finishing time

```

```

for i = 1:n
    if label(i, 2) == 1
        if ~in_interval_1
            start_time_1 = TIME_data(i);
            in_interval_1 = true;
        end
        end_time_1 = TIME_data(i);
    else
        if in_interval_1
            plot([start_time_1, end_time_1], [0, 0], 'r-', 'LineWidth', 2, 'DisplayName', '1');
            if ~isnan(last_end_time)
                plot([last_end_time, start_time_1], [last_y, 0], 'k-', 'LineWidth', 1);
            end
            last_y = 0;
            last_end_time = end_time_1;
            in_interval_1 = false;
        end
    end

    if label(i, 2) == 2
        if ~in_interval_2

            start_time_2 = TIME_data(i);
            in_interval_2 = true;

        end

        zn = ZN_valid(i);
        zp = ZP_valid(i+1);
        if (TIME_data(zp) - TIME_data(zn)) >= 3
            continue; % do not consider turns
        end
        interval_signal = GYRO_data_y(zn:zp);
        interval_time = TIME_data(zn:zp);

        % cumulative area using cumtrapz
        % the total saved area value is the last element of cumulative_area
        cumulative_area = cumtrapz(interval_time, abs(interval_signal));
        total_area = cumulative_area(end);
        % cumulative area for stance phase
        CSAVTS = [CSAVTS; total_area];
        target_area_HR = 0.46 * total_area;
        target_area_TO = 0.95 * total_area;
    end
end

```

```

% HR and TO indices using target areas
CF=round(HR_Correction_Factor(int(i))*SR);
HR_index = find(cumulative_area >= target_area_HR, 1) + zn - 1 - CF;
%return only the first index where the condition is true
%first index where the cumulative area is greater than or equal to
the target area
TO_index = find(cumulative_area >= target_area_TO, 1) + zn - 1;
% HR and TO points and the corresponding times
HR_points = [HR_points; GYRO_data_y(HR_index)];
HR_times = [HR_times; TIME_data(HR_index)];
TO_points = [TO_points; GYRO_data_y(TO_index)];
TO_times = [TO_times; TIME_data(TO_index)];
end

% cumulative area for each interval between ZP_valid and the next
ZN_valid
CSAVSW = [];
FA_points = [];
FA_times = [];
TBV_points = [];
TBV_times = [];

for i = 1:length(ZN_valid)-1
    zn = ZN_valid(i+1);

figure;
plot(TIME_data, GYRO_data_y, 'DisplayName', 'GYRO_data_y');
hold on;
plot(TIME_data(ZP_valid), zeros(size(ZP_valid)), 'rv', 'MarkerSize', 5,
'MarkerFaceColor', 'r', 'DisplayName', 'ZP');
plot(TIME_data(ZN_valid), zeros(size(ZN_valid)), 'gv', 'MarkerSize', 5,
'MarkerFaceColor', 'g', 'DisplayName', 'ZN');
plot(IC_times, GYRO_data_y(IC), 'ko', 'MarkerSize', 4, 'DisplayName',
'IC (initial contact)');
plot(HR_times, HR_points, 'mo', 'MarkerSize', 6, 'MarkerFaceColor', 'm',
'DisplayName', 'HR (heel rise)');
plot(TO_times, TO_points, 'bo', 'MarkerSize', 6, 'MarkerFaceColor', 'b',
'DisplayName', 'TO (toe off)');
plot(TBV_times, TBV_points, 'yo', 'MarkerSize', 6, 'MarkerFaceColor',
'y', 'DisplayName', 'TBV (mid-swing)');
plot(FA_times, FA_points, 'ro', 'MarkerSize', 6, 'MarkerFaceColor', 'r',
'DisplayName', 'FA (initial swing)');
xlabel('Tempo');
ylabel('GYRO_data_y');
legend('show');

```

```

% label vector initialized with NaN
label = NaN(length(TIME_data), 2);
label(:, 1) = TIME_data;
label(:, 3) = 1;
for i = 1:length(IC)
    if i <= length(HR_times)
        % IC-HR
        start_index = find(TIME_data >= TIME_data(IC(i)), 1);
        end_index = find(TIME_data < HR_times(i), 1, 'last');
        if ~isempty(start_index) && ~isempty(end_index) && HR_times(i) ->
TIME_data(IC(i)) <= 3
            label(start_index:end_index, 2) = 1;
        end
    end
    if i <= length(HR_times) && i <= length(TO_times)
        %HR-TO
        start_index = find(TIME_data >= HR_times(i), 1);
        end_index = find(TIME_data < TO_times(i), 1, 'last');

        end
    end

    end_time_2 = TIME_data(i);
    else
        if in_interval_2
            plot([start_time_2, end_time_2], [5, 5], 'b-', 'LineWidth', >
2, 'DisplayName', '2');
            if ~isnan(last_end_time)
                plot([last_end_time, start_time_2], [last_y, 5], 'b-', >
'LineWidth', 1);
            end
            last_y = 5;
            last_end_time = end_time_2;
            in_interval_2 = false;
        end
    end

    if label(i, 2) == 3
        if ~in_interval_3
            start_time_3 = TIME_data(i);
            in_interval_3 = true;
        end

        end_time_3 = TIME_data(i);
    else
        if in_interval_3
            plot([start_time_3, end_time_3], [10, 10], 'g-', >
'LineWidth', 2, 'DisplayName', '3');
            if ~isnan(last_end_time)
                plot([last_end_time, start_time_3], [last_y, 10], 'g-', >
'LineWidth', 1);
            end
            last_y = 10;
            last_end_time = end_time_3;

            in_interval_3 = false;
        end
    end

    if label(i, 2) == 4

```

```

        if ~in_interval_4
            start_time_4 = TIME_data(i);
            in_interval_4 = true;
        end

        end_time_4 = TIME_data(i);
    else
        if in_interval_4

            plot([start_time_4, end_time_4], [15, 15], 'm-', ↵
'LineWidth', 2, 'DisplayName', '4');

            if ~isnan(last_end_time)
                plot([last_end_time, start_time_4], [last_y, 15], 'm-', ↵
'LineWidth', 1);
            end

            last_y = 15;
            last_end_time = end_time_4;

            in_interval_4 = false;
        end
    end

    if label(i, 2) == 5
        if ~in_interval_5

            start_time_5 = TIME_data(i);
            in_interval_5 = true;
        end

        end_time_5 = TIME_data(i);
    else
        if in_interval_5

            plot([start_time_5, end_time_5], [20, 20], 'c-', ↵
'LineWidth', 2, 'DisplayName', '5');

            if ~isnan(last_end_time)
                plot([last_end_time, start_time_5], [last_y, 20], 'c-', ↵
'LineWidth', 1);
            end

            last_y = 20;
            last_end_time = end_time_5;

            in_interval_5 = false;
        end
    end
end
end

```

```
if in_interval_1
    plot([start_time_1, end_time_1], [0, 0], 'c-', 'LineWidth', 2);
    if ~isnan(last_y)
        plot([end_time_1, end_time_1], [last_y, 0], 'c-', 'LineWidth', 2);
    end
end

if in_interval_2
    plot([start_time_2, end_time_2], [5, 5], 'b-', 'LineWidth', 2);
end

if in_interval_3
    plot([start_time_3, end_time_3], [10, 10], 'g-', 'LineWidth', 2);
end

if in_interval_4
    plot([start_time_4, end_time_4], [15, 15], 'm-', 'LineWidth', 2);
end

if in_interval_5
    plot([start_time_5, end_time_5], [20, 20], 'c-', 'LineWidth', 2);
end
```

## Codice Matlab Segmentation\_RAMP.m

```
% This code processes gyroscope data (GYRO_data_y) and identifies
various
% gait cycle events:
% Heel Strike (HS), Toe Off (TO), Mid-Swing (MS)
% A vector called label is created where the first column contains the
time,
% the second column assigns 1 for the phase between HR to TO, 2 for TO
to MS, 3 for MS to HS.
% The third column always contains 1, indicating the group of movements
according to the methodology used.
% For each subject it is possible to visualize each trial.

subjectID = input('Enter Subject ID (01 to 20): ', 's');
trialNum = input('Enter the trial number (e.g. 01): ', 's');
% change with your folder (raw data)
load(sprintf('%s\Sub%s\Data\Sub%s_RAMP_%s.mat', 'C:\
\Users\39346\Desktop\Dataset', subjectID, subjectID, trialNum));

if length(ACC_data) ~= length(GYRO_data) || length(ACC_data) ~= length
(EMG_data) || length(GYRO_data) ~= length(EMG_data)
    % Resample
    ACC_data = [interp(ACC_data(:, 1), 4), interp(ACC_data(:, 2), 4),
interp(ACC_data(:, 3), 4)];
    GYRO_data = [interp(GYRO_data(:, 1), 4), interp(GYRO_data(:, 2), 4),
interp(GYRO_data(:, 3), 4)];
    min_length = min([length(ACC_data), length(GYRO_data), length
(EMG_data)]);
    ACC_data = ACC_data(1:min_length, :);
    GYRO_data = GYRO_data(1:min_length, :);
    EMG_data = EMG_data(1:min_length, :);
    TIME_data = TIME_data(1:min_length, :);
end

GYRO_data_y = -GYRO_data(:, 2);
TIME_data = (0:0.005:(size(EMG_data(:, 1), 1) - 1) * 0.005)';
Fs = 200;
Fc = 5; % cut off frequency

Wn = Fc / (Fs / 2);
% Second order Butterworth
[b, a] = butter(2, Wn, 'low');
GYRO_data_y = filtfilt(b, a, GYRO_data_y);
% zero-crossings
ZP = findZeroCrossings(GYRO_data_y, 'positive');
ZN = findZeroCrossings(GYRO_data_y, 'negative');
% Initialize valid zero-crossings
ZP_valid = [];
ZN_valid = [];
```

```

% Threshold for signal segmentation
th=max(GYRO_data_y)/2;
for i=1:length(ZN)
    if ZN(1) < ZP(1) && i == 1
        % If the first point found is a ZN, skip the first iteration
        continue;
    elseif ZN(1) < ZP(1)
        % If ZN appears before ZP, use GYRO_data_y(ZP(i-1):ZN(i))
        segnale = GYRO_data_y(ZP(i-1):ZN(i));
    else
        % Otherwise, use GYRO_data_y(ZP(i):ZN(i))
        segnale = GYRO_data_y(ZP(i):ZN(i));
    end
    % Check if maximum value of segment exceeds the threshold
    if max(segnale) >= th
        if ZN(1) < ZP(1) && i > 1
            ZP_valid = [ZP_valid; ZP(i-1)];
            ZN_valid = [ZN_valid; ZN(i)];
        else
            ZP_valid = [ZP_valid; ZP(i)];
            ZN_valid = [ZN_valid; ZN(i)];
        end
    end
end
end

% Remove ZN and ZP if the interval between them is greater than 2✓
seconds
i = 1;
while i < length(ZN_valid)
    if TIME_data(ZP_valid(i+1)) - TIME_data(ZN_valid(i)) > 2
        % Remove ZN and ZP

        ZN_valid(i:i+1) = [];
        ZP_valid(i:i+1) = [];
        i = i + 1;
    else

        i = i + 1;
    end
end

% Remove the last ZP and ZN
ZP_valid(end) = [];
ZN_valid(end) = [];
% derivate of GYRO_data_y
GYRO_data_y_diff = diff(GYRO_data_y); %points where the signal changes
%first minimum after each ZN_valid point
HS = [];
for i = 1:length(ZN_valid)
    zn = ZN_valid(i);
    for j = zn:length(GYRO_data_y_diff)-1
        if GYRO_data_y_diff(j) < 0 && GYRO_data_y_diff(j + 1) >= 0
            HS = [HS; j+1];
            break;
        end
    end
end
end
end

```

```

TO = [];
for i = 1:length(ZP_valid)
    zp = ZP_valid(i);
    for j = zp-1:-1:1
        if GYRO_data_y_diff(j) < 0 && GYRO_data_y_diff(j + 1) >= 0
            TO = [TO; j];
            break;
        end
    end
end

MS = [];
for i = 1:length(ZP_valid)
    if i < length(ZN_valid)
        segment = GYRO_data_y(ZP_valid(i):ZN_valid(i));

        [peaks, peak_idx] = findpeaks(segment);
        if ~isempty(peak_idx)
            MS = [MS; ZP_valid(i) + peak_idx - 1];
        end
    end
end

if ~isempty(MS) && MS(1) < HS(1)
    MS = MS(MS > HS(1));
end

if ~isempty(TO) && TO(1) < HS(1)
    TO = TO(TO > HS(1));
end

figure;
plot(TIME_data, GYRO_data_y, 'DisplayName', 'GYRO');
hold on;
plot(TIME_data([ZP_valid; ZN_valid]), zeros(size([ZP_valid; ZN_valid])),
...
    'rv', 'MarkerSize', 3, 'MarkerFaceColor', 'r', 'DisplayName',
'ZC');
plot(TIME_data(HS), GYRO_data_y(HS), 'go', 'MarkerSize', 4,
'MarkerFaceColor', 'g', 'DisplayName', 'HS (Heel Strike)');
plot(TIME_data(TO), GYRO_data_y(TO), 'bo', 'MarkerSize', 4,
'MarkerFaceColor', 'b', 'DisplayName', 'TO (Toe Off)');
plot(TIME_data(MS), GYRO_data_y(MS), 'yo', 'MarkerSize', 4,
'MarkerFaceColor', 'y', 'DisplayName', 'MS (Mid-Swing)');
xlabel('Time [s]');
ylabel('Angular rate [rad/s]');
legend('show');

% label vector initialized with NaN
label = NaN(length(TIME_data), 2);
label(:, 1) = TIME_data;
label(:, 3) = 1;
for i = 1:length(HS)
    if i <= length(TO) && i <= length(MS)
        % HS-TO
        start_index = find(TIME_data >= TIME_data(HS(i)), 1);
        end_index = find(TIME_data < TIME_data(TO(i)), 1, 'last');
        if ~isempty(start_index) && ~isempty(end_index) && (TIME_data(TO
(i)) - TIME_data(HS(i)) <= 3)

```

```

        label(start_index:end_index, 2) = 1;
    end

    % TO-MS
    start_index = find(TIME_data >= TIME_data(TO(i)), 1);
    end_index = find(TIME_data < TIME_data(MS(i)), 1, 'last');
    if ~isempty(start_index) && ~isempty(end_index) && (TIME_data(MS
(i)) - TIME_data(TO(i)) <= 3)
        label(start_index:end_index, 2) = 2;
    end

    % MS-HS
    if i < length(HS)
        start_index = find(TIME_data >= TIME_data(MS(i)), 1);
        end_index = find(TIME_data < TIME_data(HS(i+1)), 1,
'last');
        if ~isempty(start_index) && ~isempty(end_index) &&
(TIME_data(HS(i+1)) - TIME_data(MS(i)) <= 3)
            label(start_index:end_index, 2) = 3;
        end
    end
end
end
end

max_GYRO = max(abs(GYRO_data_y));
GYRO_data_y_normalized = GYRO_data_y / max_GYRO * 20; % Normalizes the
data so that the maximum value is 20
HS_points_normalized = GYRO_data_y(HS) / max_GYRO * 20;
TO_points_normalized = GYRO_data_y(TO) / max_GYRO * 20;
MS_points_normalized = GYRO_data_y(MS) / max_GYRO * 20;

figure;
plot(TIME_data, GYRO_data_y_normalized, 'DisplayName', 'GYRO_data_y');
hold on;
plot(TIME_data(ZP_valid), zeros(size(ZP_valid)), 'ro', 'MarkerSize', 4);
plot(TIME_data(ZN_valid), zeros(size(ZN_valid)), 'ro', 'MarkerSize', 4);
plot(TIME_data(HS), HS_points_normalized, 'ko', 'MarkerSize', 4);
plot(TIME_data(TO), TO_points_normalized, 'mo', 'MarkerSize', 6);
plot(TIME_data(MS), MS_points_normalized, 'bo', 'MarkerSize', 6);

n = size(label, 1);
start_time_1 = [];
end_time_1 = [];
start_time_2 = [];
end_time_2 = [];
start_time_3 = [];
end_time_3 = [];
in_interval_1 = false;
in_interval_2 = false;
in_interval_3 = false;

last_y = NaN; % To keep track of the latest y value
last_end_time = NaN; % To track the latest finishing time

```

```

for i = 1:n

    if label(i, 2) == 1
        if ~in_interval_1

            start_time_1 = TIME_data(i);
            in_interval_1 = true;
            end

            end_time_1 = TIME_data(i);
        else
            if in_interval_1

                plot([start_time_1, end_time_1], [0, 0], 'r-', 'LineWidth', ↵
2, 'DisplayName', '1');

                if ~isnan(last_end_time)
                    plot([last_end_time, start_time_1], [last_y, 0], 'r-', ↵
'LineWidth', 1);
                end
                last_y = 0;
                last_end_time = end_time_1;

                in_interval_1 = false;
            end
        end
    end

    if label(i, 2) == 2
        if ~in_interval_2

            start_time_2 = TIME_data(i);
            in_interval_2 = true;
            end

            end_time_2 = TIME_data(i);
        else
            if in_interval_2

                plot([start_time_2, end_time_2], [5, 5], 'b-', 'LineWidth', ↵
2, 'DisplayName', '2');
                if ~isnan(last_end_time)
                    plot([last_end_time, start_time_2], [last_y, 5], 'b-', ↵
'LineWidth', 1);
                end

                last_y = 5;
                last_end_time = end_time_2;

                in_interval_2 = false;
            end
        end
    end
end

```

```

if label(i, 2) == 3
    if ~in_interval_3
        start_time_3 = TIME_data(i);
        in_interval_3 = true;
    end

    end_time_3 = TIME_data(i);
else
    if in_interval_3
        plot([start_time_3, end_time_3], [10, 10], 'g-', ↵
'LineWidth', 2, 'DisplayName', '3');

        if ~isnan(last_end_time)
            plot([last_end_time, start_time_3], [last_y, 10], 'g-', ↵
'LineWidth', 1);
        end

        last_y = 10;
        last_end_time = end_time_3;

        in_interval_3 = false;
    end
end
end

if in_interval_1
    plot([start_time_1, end_time_1], [0, 0], 'c-', 'LineWidth', 2);
    if ~isnan(last_y)
        plot([end_time_1, end_time_1], [last_y, 0], 'c-', 'LineWidth', ↵
1);
    end
end

if in_interval_2
    plot([start_time_2, end_time_2], [5, 5], 'b-', 'LineWidth', 2);
end

if in_interval_3
    plot([start_time_3, end_time_3], [10, 10], 'g-', 'LineWidth', 2);
end

```

## Codice Matlab Segmentation\_STAIR.m

```
subjectID = input('Enter Subject ID (01 to 20): ', 's');
trialNum = input('Enter the trial number (e.g. 01): ', 's');

% This code processes gyroscope data (GYRO_data_y) and identifies
various
% gait cycle events:
% Initial contact (IC), Foot Off (FO)
% A vector called label is created where the first column contains the
time,
% the second column assigns 1 for the phase between IC to FO, 2 for FO
to IC
% The third column always contains 1, indicating the group of movements
according to the methodology used.
% For each subject it is possible to visualize each trial.

% change with the your Dataset folder (RAW Data)
load(sprintf('%s\\Sub%s\\Data\\Sub%s_STAIR_%s.mat', 'C:\
\Users\39346\Desktop\Dataset', subjectID, subjectID, trialNum));

if length(ACC_data) ~= length(GYRO_data) || length(ACC_data) ~= length
(EMG_data) || length(GYRO_data) ~= length(EMG_data)
    % Resample
    ACC_data = [interp(ACC_data(:, 1), 4), interp(ACC_data(:, 2), 4),
interp(ACC_data(:, 3), 4)];
    GYRO_data = [interp(GYRO_data(:, 1), 4), interp(GYRO_data(:, 2), 4),
interp(GYRO_data(:, 3), 4)];
    % Synchronization of data lengths
    min_length = min([length(ACC_data), length(GYRO_data), length
(EMG_data)]);
    ACC_data = ACC_data(1:min_length, :);
    GYRO_data = GYRO_data(1:min_length, :);
    EMG_data = EMG_data(1:min_length, :);
    TIME_data = TIME_data(1:min_length, :);
end

GYRO_data_y = -GYRO_data(:, 2);
TIME_data = (0:0.005:(size(EMG_data(:, 1), 1) - 1) * 0.005)';
Fs = 200;

Fc = 10;    % cut off frequency
Wn = Fc / (Fs / 2);

% Second order Butterworth filter
[b, a] = butter(2, Wn, 'low');
GYRO_data_y = filtfilt(b, a, GYRO_data_y);
% zero-crossings
ZP = findZeroCrossings(GYRO_data_y, 'positive');
ZN = findZeroCrossings(GYRO_data_y, 'negative');
% Initialize valid zero-crossings
ZP_valid = [];
ZN_valid = [];
```

```

% Threshold for signal segmentation
th=max(GYRO_data_y)/3;
for i=1:length(ZN)
    if ZN(1) < ZP(1) && i == 1
        % If the first point found is a ZN, skip the first iteration
        continue;
    elseif ZN(1) < ZP(1)
        % If ZN appears before ZP, use GYRO_data_y(ZP(i-1):ZN(i))
        segnale = GYRO_data_y(ZP(i-1):ZN(i));
    else
        % Otherwise, use GYRO_data_y(ZP(i):ZN(i))
        segnale = GYRO_data_y(ZP(i):ZN(i));
    end
    % Check if maximum value of segment exceeds the threshold
    if max(segnale) >= th
        if ZN(1) < ZP(1) && i > 1
            ZP_valid = [ZP_valid; ZP(i-1)];
            ZN_valid = [ZN_valid; ZN(i)];
        else
            ZP_valid = [ZP_valid; ZP(i)];
            ZN_valid = [ZN_valid; ZN(i)];
        end
    end
end
end

% Remove ZN and ZP if the interval between them is greater than 1.5 seconds
i = 1;
while i < length(ZN_valid)
    if TIME_data(ZP_valid(i+1)) - TIME_data(ZN_valid(i)) > 1.5
        % Remove ZN and ZP
        ZN_valid(i:i+1) = [];
        ZP_valid(i:i+1) = [];

        i = i + 1;
    else
        i = i + 1;
    end
end

% Remove the last ZP and ZN
ZP_valid(end-1:end) = [];
ZN_valid(end-1:end) = [];
% derivate of GYRO_data_y
GYRO_data_y_diff = diff(GYRO_data_y); %points where the signal changes
%first minimum after each ZN_valid point
% Identify initial contact points (IC)
IC = [];
for i = 1:length(ZN_valid)
    zn = ZN_valid(i);
    for j = zn:length(GYRO_data_y_diff)-1
        if GYRO_data_y_diff(j) < 0 && GYRO_data_y_diff(j + 1) >= 0
            IC = [IC; j+1];
            break;
        end
    end
end
end
end

```

```

FO = [];
for i = 1:length(ZP_valid)
    zp = ZP_valid(i);
    for j = zp-1:-1:1
        if GYRO_data_y_diff(j) < 0 && GYRO_data_y_diff(j + 1) >= 0
            FO = [FO; j];
            break;
        end
    end
end

if ~isempty(FO) && FO(1) < IC(1)
    FO = FO(FO > IC(1));
end

figure;
plot(TIME_data, GYRO_data_y, 'DisplayName', 'GYRO_data_y');
hold on;
plot(TIME_data([ZP_valid; ZN_valid]), zeros(size([ZP_valid; ZN_valid])),
...
    'rv', 'MarkerSize', 3, 'MarkerFaceColor', 'r', 'DisplayName',
'ZC');
plot(TIME_data(IC), GYRO_data_y(IC), 'go', 'MarkerSize', 4,
'MarkerFaceColor', 'g', 'DisplayName', 'IC (Initial Contact)');
plot(TIME_data(FO), GYRO_data_y(FO), 'bo', 'MarkerSize', 4,
'MarkerFaceColor', 'b', 'DisplayName', 'FO (Foot Off)');
xlabel('Time [s]');
ylabel('Angular rate [rad/s]');
legend('show');
% label vector initialized with NaN
label = NaN(length(TIME_data), 2);
label(:, 1) = TIME_data;
label(:, 3) = 1;
for i = 1:length(IC)
    if i <= length(FO)
        % IC-FO
        start_index = find(TIME_data >= TIME_data(IC(i)), 1);
        end_index = find(TIME_data < TIME_data(FO(i)), 1, 'last');
        if ~isempty(start_index) && ~isempty(end_index) && (TIME_data(FO
(i)) - TIME_data(IC(i)) <= 2)
            label(start_index:end_index, 2) = 1;
        end

        %FO-IC
        if i < length(IC)
            start_index = find(TIME_data >= TIME_data(FO(i)), 1);
            end_index = find(TIME_data < TIME_data(IC(i+1)), 1,
'last');
            if ~isempty(start_index) && ~isempty(end_index) &&
(TIME_data(IC(i+1)) - TIME_data(FO(i)) <= 2)
                label(start_index:end_index, 2) = 2;
            end
        end
    end
end

```

```

        end
    end
end
end

max_GYRO = max(abs(GYRO_data_y));
GYRO_data_y_normalized = GYRO_data_y / max_GYRO * 20; % Normalizes the
data so that the maximum value is 20
IC_points_normalized = GYRO_data_y(IC) / max_GYRO * 20;
FO_points_normalized = GYRO_data_y(FO) / max_GYRO * 20;
figure;
plot(TIME_data, GYRO_data_y_normalized, 'DisplayName', 'GYRO_data_y');
hold on;
plot(TIME_data(ZP_valid), zeros(size(ZP_valid)), 'ro', 'MarkerSize', 4);
plot(TIME_data(ZN_valid), zeros(size(ZN_valid)), 'ro', 'MarkerSize', 4);
plot(TIME_data(IC), IC_points_normalized, 'ko', 'MarkerSize', 4);
plot(TIME_data(FO), FO_points_normalized, 'mo', 'MarkerSize', 6);
n = size(label, 1);
start_time_1 = [];
end_time_1 = [];
start_time_2 = [];
end_time_2 = [];
in_interval_1 = false;
in_interval_2 = false;
last_y = NaN; % To keep track of the latest y value
last_end_time = NaN; % To track the latest finishing time
for i = 1:n
    if label(i, 2) == 1
        if ~in_interval_1

            start_time_1 = TIME_data(i);
            in_interval_1 = true;

            end_time_1 = TIME_data(i);
        else
            if in_interval_1

                plot([start_time_1, end_time_1], [0, 0], 'r-', 'LineWidth', 2, 'DisplayName', '1');

                if ~isnan(last_end_time)
                    plot([last_end_time, start_time_1], [last_y, 0], 'r-', 'LineWidth', 1);
                end

                last_y = 0;
                last_end_time = end_time_1;

                in_interval_1 = false;
            end
        end
    if label(i, 2) == 2
        if ~in_interval_2

            start_time_2 = TIME_data(i);
            in_interval_2 = true;

            end
        end
    end
end

```

```

        end_time_2 = TIME_data(i);
    else
        if in_interval_2

            plot([start_time_2, end_time_2], [5, 5], 'b-', 'LineWidth', 2, 'DisplayName', '2');

            if ~isnan(last_end_time)
                plot([last_end_time, start_time_2], [last_y, 5], 'b-', 'LineWidth', 1);
            end
            last_y = 5;
            last_end_time = end_time_2;

            in_interval_2 = false;
        end
    end

end

if in_interval_1
    plot([start_time_1, end_time_1], [0, 0], 'c-', 'LineWidth', 2);

    if ~isnan(last_y)
        plot([end_time_1, end_time_1], [last_y, 0], 'c-', 'LineWidth', 1);
    end
end

if in_interval_2
    plot([start_time_2, end_time_2], [5, 5], 'b-', 'LineWidth', 2);
end

```

## Codice Matlab SaveLabel\_WAK.m

```
% A vector named 'label' is created, which contains:
% - The first column with time data.
% - The second column with phase labels: 1 for the IC-HR phase, 2 for
the HR-TO phase, 3 for the TO-FA phase, 4 for the FA-TBV phase, and 5
for the TBV-IC phase.
% - The third column is a constant value of 1, representing the group of
movements.
% The results are then saved to an Excel file

maxTrials = 8;
% Change with the path of the folder where the data is taken from
baseFolder = 'C:\Users\39346\Desktop\Dataset';
% Change with the destination folder path
destFolder = 'C:\Users\39346\Desktop\DatasetFile';

for subject = 1:20
    subjectField = sprintf('Sub%02d', subject);
    % Destination folder
    subjectDestFolder = fullfile(destFolder, subjectField, 'Label');
    if ~exist(subjectDestFolder, 'dir')
        mkdir(subjectDestFolder);
    end
    for trial = 1:maxTrials
        fileName = sprintf('%s/Sub%02d/Data/Sub%02d_WAK_%02d.mat',
baseFolder, subject, subject, trial);
        if exist(fileName, 'file')
            load(fileName, 'ACC_data', 'GYRO_data', 'EMG_data',
'TIME_data');
            if length(ACC_data) ~= length(GYRO_data) || length(ACC_data) ~= length
(EMG_data) || length(GYRO_data) ~= length(EMG_data)
                % Resample
                ACC_data = [interp(ACC_data(:, 1), 4), interp(ACC_data(:, 2), 4), interp
(ACC_data(:, 3), 4)];
                GYRO_data = [interp(GYRO_data(:, 1), 4), interp(GYRO_data(:, 2), 4),
interp(GYRO_data(:, 3), 4)];
                % Synchronization of data lengths
                min_length = min([length(ACC_data), length(GYRO_data), length
(EMG_data)]);
                ACC_data = ACC_data(1:min_length, :);
                GYRO_data = GYRO_data(1:min_length, :);
            end
        end
    end
end
```

```

EMG_data = EMG_data(1:min_length, :);
TIME_data = TIME_data(1:min_length, :);
end
TIME_data=(0:0.005:(size(EMG_data(:,1),1)-1)*0.005)'; %rebuilding the
correct time vector
GYRO_data_y= -GYRO_data(:,2);
SR=200;
%Correction_factors
HR_Correction_Factor = @(stride_time) 0.156 - 0.154 * stride_time;
FA_Correction_Factor = @(ZPcp) -0.254 + 0.384 * ZPcp;

%zero-crossings
ZP = findZeroCrossings(GYRO_data_y, 'positive');
ZN = findZeroCrossings(GYRO_data_y, 'negative');
ZP_valid = [];
ZN_valid = [];

th=max(GYRO_data_y)/2;
for i=1:length(ZN)
    if ZN(1) < ZP(1) && i == 1
        % If the first point found is a ZN, skip the first iteration
        continue;
    elseif ZN(1) < ZP(1)
        % If ZN appears before ZP, use GYRO_data_y(ZP(i-1):ZN(i))
        segnale = GYRO_data_y(ZP(i-1):ZN(i));
    else
        % Otherwise, use GYRO_data_y(ZP(i):ZN(i))
        segnale = GYRO_data_y(ZP(i):ZN(i));
    end
    if max(segnale) >= th
        if ZN(1) < ZP(1) && i > 1
            ZP_valid = [ZP_valid; ZP(i-1)];
            ZN_valid = [ZN_valid; ZN(i)];
        else
            ZP_valid = [ZP_valid; ZP(i)];
            ZN_valid = [ZN_valid; ZN(i)];
        end
    end
end
end

% Remove ZN and ZP if the interval between them is greater than 3
seconds
i = 1;
while i < length(ZN_valid)
    if TIME_data(ZP_valid(i+1)) - TIME_data(ZN_valid(i)) > 3
        % Remove ZN and ZP
        ZN_valid(i:i+1) = [];
        ZP_valid(i:i+1) = [];
        i = i + 1;
    else
        i = i + 1;
    end
end
end

```

```

% Remove the last ZP and ZN
ZP_valid(end) = [];
ZN_valid(end) = [];

% derivate of GYRO_data_y
GYRO_data_y_diff = diff(GYRO_data_y); %points where the signal changes
%first minimum after each ZN_valid point
IC = [];
for i = 1:length(ZN_valid)
    zn = ZN_valid(i);
    for j = zn:length(GYRO_data_y_diff)-1
        if GYRO_data_y_diff(j) < 0 && GYRO_data_y_diff(j + 1) >= 0
            IC = [IC; j+1];
            break;
        end
    end
end
end

int = TIME_data(IC(2:end)) - TIME_data(IC(1:end-1));
% cumulative area for each interval between ZN_valid and the next ✓
ZP_valid
CSAVTS = [];
HR_points = [];
HR_times = [];
TO_points = [];
TO_times = [];

for i = 1:length(ZN_valid)-1
    zn = ZN_valid(i);
    zp = ZP_valid(i+1);
    if (TIME_data(zp) - TIME_data(zn)) >= 3
        continue; % do not consider turns
    end
    interval_signal = GYRO_data_y(zn:zp);
    interval_time = TIME_data(zn:zp);

    % cumulative area using cumtrapz
    % the total saved area value is the last element of cumulative_area
    cumulative_area = cumtrapz(interval_time, abs(interval_signal));
    total_area = cumulative_area(end);
    % cumulative area for stance phase
    CSAVTS = [CSAVTS; total_area];
    target_area_HR = 0.46 * total_area;
    target_area_TO = 0.95 * total_area;
end

```

```

% HR and TO indices using target areas
CF=round(HR_Correction_Factor(int(i))*SR);
HR_index = find(cumulative_area >= target_area_HR, 1) + zn - 1 - CF;
%return only the first index where the condition is true
%first index where the cumulative area is greater than or equal to
the target area
TO_index = find(cumulative_area >= target_area_TO, 1) + zn - 1;
% HR and TO points and the corresponding times
HR_points = [HR_points; GYRO_data_y(HR_index)];
HR_times = [HR_times; TIME_data(HR_index)];
TO_points = [TO_points; GYRO_data_y(TO_index)];
TO_times = [TO_times; TIME_data(TO_index)];
end

% cumulative area for each interval between ZP_valid and the next
ZN_valid
CSAVSW = [];
FA_points = [];
FA_times = [];
TBV_points = [];
TBV_times = [];

for i = 1:length(ZN_valid)-1
    zn = ZN_valid(i+1);
    zp = ZP_valid(i+1);
    zpcp=(zp-IC(i))/(IC(i+1)-IC(i));
    CF=round(FA_Correction_Factor(zpcp)*SR);
    if (TIME_data(zn) - TIME_data(zp)) >= 3
        continue;
    end
    interval_signal = GYRO_data_y(zp:zn);
    interval_time = TIME_data(zp:zn);
    cumulative_area = cumtrapz(interval_time, abs(interval_signal));
    total_area = cumulative_area(end);

    % cumulative area for swing phase
    CSAVSW = [CSAVSW; total_area];
    target_area_FA = 0.20 * total_area;
    target_area_TBV = 0.731 * total_area;
    FA_index = find(cumulative_area >= target_area_FA & GYRO_data_y(zp:
zn) > 100, 1) + zp - 1 - CF;
    TBV_index = find(cumulative_area >= target_area_TBV & GYRO_data_y
(zp:zn) > 100, 1) + zp - 1;

    % Verify that the FA and TBV points are above 100 on the y-axis
    if ~isempty(FA_index) && ~isempty(TBV_index) && GYRO_data_y
(FA_index) > 100 && GYRO_data_y(TBV_index) > 100
        FA_points = [FA_points; GYRO_data_y(FA_index)];
        FA_times = [FA_times; TIME_data(FA_index)];
        TBV_points = [TBV_points; GYRO_data_y(TBV_index)];
        TBV_times = [TBV_times; TIME_data(TBV_index)];
    end
end
end

```

```

% Correction of IC points
IC_diff_times = diff(TIME_data(IC));
valid_IC_indices = true(length(IC), 1);
for i = 1:length(IC_diff_times)
    if IC_diff_times(i) > 5
        % If the time difference is more than 5 seconds, delete the
first IC
        valid_IC_indices(i) = false;
    end
end

end

valid_IC_indices(1) = true;
IC = IC(valid_IC_indices);
IC_times = TIME_data(IC);
% label vector initialized with NaN
label = NaN(length(TIME_data), 2);
label(:, 1) = TIME_data;
label(:, 3) = 1;
for i = 1:length(IC)
    if i <= length(HR_times)
        % IC-HR
        start_index = find(TIME_data >= TIME_data(IC(i)), 1);
        end_index = find(TIME_data < HR_times(i), 1, 'last');
        if ~isempty(start_index) && ~isempty(end_index) && HR_times(i) -
TIME_data(IC(i)) <= 3
            label(start_index:end_index, 2) = 1;
        end
    end

    if i <= length(HR_times) && i <= length(TO_times)
        % HR-TO
        start_index = find(TIME_data >= HR_times(i), 1);
        end_index = find(TIME_data < TO_times(i), 1, 'last');
        if ~isempty(start_index) && ~isempty(end_index) && TO_times(i) -
HR_times(i) <= 3
            label(start_index:end_index, 2) = 2;
        end
    end

    if i <= length(TO_times) && i <= length(FA_times)
        % TO-FA
        start_index = find(TIME_data >= TO_times(i), 1);
        end_index = find(TIME_data < FA_times(i), 1, 'last');
        if ~isempty(start_index) && ~isempty(end_index) && FA_times(i) -
TO_times(i) <= 3
            label(start_index:end_index, 2) = 3;
        end
    end

    if i <= length(FA_times) && i <= length(TBV_times)

```

```

        % FA-TBV
        start_index = find(TIME_data >= FA_times(i), 1);
        end_index = find(TIME_data < TBV_times(i), 1, 'last');
        if ~isempty(start_index) && ~isempty(end_index) && TBV_times(i) ✓
- FA_times(i) <= 3
            label(start_index:end_index, 2) = 4;
        end
    end

    if i <= length(TBV_times) && i < length(IC)
        %TBV-IC
        start_index = find(TIME_data >= TBV_times(i), 1);
        end_index = find(TIME_data < TIME_data(IC(i + 1)), 1, 'last');
        if ~isempty(start_index) && ~isempty(end_index) && TIME_data(IC ✓
(i + 1)) - TBV_times(i) <= 3
            label(start_index:end_index, 2) = 5;
        end
    end
end

% Manage intervals that are too long
for i = 1:length(IC)
    if i < length(HR_times)
        if HR_times(i) - TIME_data(IC(i)) > 3
            next_start = TIME_data(IC(i + 1));
            if i + 1 <= length(HR_times)
                next_end = HR_times(i + 1);
                label(TIME_data >= next_start & TIME_data < next_end, 2) ✓
= 1;
            end
        end
    end

    if i < length(HR_times) && i < length(TO_times)
        if TO_times(i) - HR_times(i) > 3
            next_start = HR_times(i + 1);
            if i + 1 <= length(TO_times)
                next_end = TO_times(i + 1);
                label(TIME_data >= next_start & TIME_data < next_end, 2) ✓
= 2;
            end
        end
    end

    if i < length(TO_times) && i < length(FA_times)
        if FA_times(i) - TO_times(i) > 3
            next_start = TO_times(i + 1);
            if i + 1 <= length(FA_times)
                next_end = FA_times(i + 1);
                label(TIME_data >= next_start & TIME_data < next_end, 2) ✓
= 3;
            end
        end
    end
end

```

```

    if i < length(FA_times) && i < length(TBV_times)
        if TBV_times(i) - FA_times(i) > 3
            next_start = FA_times(i + 1);
            if i + 1 <= length(TBV_times)
                next_end = TBV_times(i + 1);
                label(TIME_data >= next_start & TIME_data < next_end, 2) ✓
            = 4;
            end
        end
    end

    if i < length(TBV_times) && i < length(IC)
        if TIME_data(IC(i + 1)) - TBV_times(i) > 3
            next_start = TBV_times(i + 1);
            if i + 1 < length(IC)
                next_end = TIME_data(IC(i + 1));
                label(TIME_data >= next_start & TIME_data < next_end, 2) ✓
            = 5;
            end
        end
    end

    %table of cells to handle both numbers and strings
    labelCell = num2cell(label);

    % Replace NaNs with 'NaN' (string) in column 2

    for i = 1:size(labelCell, 1)
        if isnan(labelCell{i, 2})
            labelCell{i, 2} = 'NaN';
        else
            labelCell{i, 2} = num2str(labelCell{i, 2});
        end
    end

    % Create a table
    t = cell2table(labelCell, 'VariableNames', {'Time', 'Status', 'Group'});

    % output file
    outputFile = fullfile(subjectDestFolder, sprintf('Sub%02d_WAK_%02d_Label.xls', subject, trial));

    % Save in excel file
    writetable(t, outputFile, 'FileType', 'spreadsheet');
    else
        break;
    end
end
end
end

```

## Codici Matlab SaveLabel\_RAMP.m

```
% A vector named 'label' is created, which contains:
% - The first column with time data.
% - The second column with phase labels: 1 for the HS-TO phase, 2 for
the TO-MS phase, 3 for the MS-HS phase
% - The third column is a constant value of 1, representing the group of
movements.
% The results are then saved to an Excel file
maxTrials = 7;
% Change with the path of the folder where the data is taken from
baseFolder = 'C:\Users\39346\Desktop\Dataset';
% Change with the destination folder path
destFolder = 'C:\Users\39346\Desktop\DatasetFile';
for subject = 1:20
    subjectField = sprintf('Sub%02d', subject);
    % Destination folder
    subjectDestFolder = fullfile(destFolder, subjectField, 'Label');
    if ~exist(subjectDestFolder, 'dir')
        mkdir(subjectDestFolder);
    end
    for trial = 1:maxTrials
        fileName = sprintf('%s/Sub%02d/Data/Sub%02d_RAMP_%02d.mat',
baseFolder, subject, subject, trial);
        if exist(fileName, 'file')
            load(fileName, 'ACC_data', 'GYRO_data', 'EMG_data',
'TIME_data');

if length(ACC_data) ~= length(GYRO_data) || length(ACC_data) ~= length
(EMG_data) || length(GYRO_data) ~= length(EMG_data)
% Resample
ACC_data = [interp(ACC_data(:, 1), 4), interp(ACC_data(:, 2), 4), interp
(ACC_data(:, 3), 4)];
GYRO_data= [interp(GYRO_data(:, 1), 4), interp(GYRO_data(:, 2), 4),
interp(GYRO_data(:, 3), 4)];
% Synchronization of data lengths
min_length = min([length(ACC_data), length(GYRO_data), length
(EMG_data)]);
ACC_data = ACC_data(1:min_length, :);
GYRO_data = GYRO_data(1:min_length, :);
EMG_data = EMG_data(1:min_length, :);
TIME_data = TIME_data(1:min_length, :);
end
TIME_data=(0:0.005:(size(EMG_data(:,1),1)-1)*0.005)'; %rebuilding the
correct time vector

GYRO_data_y = -GYRO_data(:, 2);
TIME_data = (0:0.005:(size(EMG_data(:, 1), 1) - 1) * 0.005)'; %
Ricostruzione del vettore temporale
Fs = 200;
Fc = 5; % cut off frequency
Wn = Fc / (Fs / 2);
% Second order Butterworth filter
[b, a] = butter(2, Wn, 'low');
GYRO_data_y = filtfilt(b, a, GYRO_data_y);
```

```

% zero-crossings
ZP = findZeroCrossings(GYRO_data_y, 'positive');
ZN = findZeroCrossings(GYRO_data_y, 'negative');

% Initialize valid zero-crossings
ZP_valid = [];
ZN_valid = [];

% Threshold for signal segmentation
th=max(GYRO_data_y)/2;

for i=1:length(ZN)
    if ZN(1) < ZP(1) && i == 1
        % If the first point found is a ZN, skip the first iteration
        continue;
    elseif ZN(1) < ZP(1)
        % If ZN appears before ZP, use GYRO_data_y(ZP(i-1):ZN(i))
        segnale = GYRO_data_y(ZP(i-1):ZN(i));
    else
        % Otherwise, use GYRO_data_y(ZP(i):ZN(i))
        segnale = GYRO_data_y(ZP(i):ZN(i));
    end

    % Check if maximum value of segment exceeds the threshold
    if max(segnale) >= th
        if ZN(1) < ZP(1) && i > 1
            ZP_valid = [ZP_valid; ZP(i-1)];
            ZN_valid = [ZN_valid; ZN(i)];
        else
            ZP_valid = [ZP_valid; ZP(i)];
            ZN_valid = [ZN_valid; ZN(i)];
        end
    end
end

% Remove ZN and ZP if the interval between them is greater than 2✓
seconds
i = 1;
while i < length(ZN_valid)
    if TIME_data(ZP_valid(i+1)) - TIME_data(ZN_valid(i)) > 2
        % Remove ZN and ZP
        ZN_valid(i:i+1) = [];
        ZP_valid(i:i+1) = [];

        i = i + 1;
    else
        i = i + 1;
    end
end

% Remove the last ZP and ZN
ZP_valid(end) = [];
ZN_valid(end) = [];

```

```

% derivate of GYRO_data_y
GYRO_data_y_diff = diff(GYRO_data_y); %points where the signal changes
%first minimum after each ZN_valid point

HS = [];
for i = 1:length(ZN_valid)
    zn = ZN_valid(i);
    for j = zn:length(GYRO_data_y_diff)-1
        if GYRO_data_y_diff(j) < 0 && GYRO_data_y_diff(j + 1) >= 0

            HS = [HS; j+1];
            break;
        end
    end
end

TO = [];
for i = 1:length(ZP_valid)
    zp = ZP_valid(i);
    for j = zp-1:-1:1
        if GYRO_data_y_diff(j) < 0 && GYRO_data_y_diff(j + 1) >= 0
            TO = [TO; j];
            break;
        end
    end
end

MS = [];
for i = 1:length(ZP_valid)
    if i < length(ZN_valid)
        segment = GYRO_data_y(ZP_valid(i):ZN_valid(i));
        [peaks, peak_idx] = findpeaks(segment);
        if ~isempty(peak_idx)
            MS = [MS; ZP_valid(i) + peak_idx - 1];
        end
    end
end

if ~isempty(MS) && MS(1) < HS(1)
    MS = MS(MS > HS(1));
end

if ~isempty(TO) && TO(1) < HS(1)
    TO = TO(TO > HS(1));
end

% label vector initialized with NaN
label = NaN(length(TIME_data), 2);
label(:, 1) = TIME_data;
label(:, 3) = 1;
for i = 1:length(HS)
    if i <= length(TO) && i <= length(MS)
        % HS-TO
        start_index = find(TIME_data >= TIME_data(HS(i)), 1);
    end
end

```

```

        end_index = find(TIME_data < TIME_data(TO(i)), 1, 'last');
        if ~isempty(start_index) && ~isempty(end_index) && (TIME_data(TO(i)) - TIME_data(HS(i)) <= 3)
            label(start_index:end_index, 2) = 1;
        end

        % TO-MS
        start_index = find(TIME_data >= TIME_data(TO(i)), 1);
        end_index = find(TIME_data < TIME_data(MS(i)), 1, 'last');
        if ~isempty(start_index) && ~isempty(end_index) && (TIME_data(MS(i)) - TIME_data(TO(i)) <= 3)
            label(start_index:end_index, 2) = 2;
        end

        % MS-HS
        if i < length(HS)
            start_index = find(TIME_data >= TIME_data(MS(i)), 1);
            end_index = find(TIME_data < TIME_data(HS(i+1)), 1, 'last');
            if ~isempty(start_index) && ~isempty(end_index) && (TIME_data(HS(i+1)) - TIME_data(MS(i)) <= 3)
                label(start_index:end_index, 2) = 3;
            end
        end
    end
end

%table of cells to handle both numbers and strings
labelCell = num2cell(label);

% Replace NaNs with 'NaN' (string) in column 2
for i = 1:size(labelCell, 1)
    if isnan(labelCell{i, 2})
        labelCell{i, 2} = 'NaN';
    else
        labelCell{i, 2} = num2str(labelCell{i, 2});
    end
end

% Create a table
t = cell2table(labelCell, 'VariableNames', {'Time', 'Status', 'Group'});

% output file
outputFile = fullfile(subjectDestFolder, sprintf('Sub%02d_RAMP_%02d_Label.xls', subject, trial));

% Save in excel file
writetable(t, outputFile, 'FileType', 'spreadsheet');
else
    break;
end
end
end

```

## Codice Matlab SaveLabel\_STAIR.m

```
% A vector named 'label' is created, which contains:
% - The first column with time data.
% - The second column with phase labels: 1 for the IC-FO phase, 2 for the
FO-IC phase
% - The third column is a constant value of 1, representing the group of
movements.
% The results are then saved to an Excel file
maxTrials = 8;
% Change with the path of the folder where the data is taken from
baseFolder = 'C:\Users\39346\Desktop\Dataset';
% Change with the destination folder path
destFolder = 'C:\Users\39346\Desktop\DatasetFile';
for subject = 1:20
    subjectField = sprintf('Sub%02d', subject);
    % Destination folder
    subjectDestFolder = fullfile(destFolder, subjectField, 'Label');
    if ~exist(subjectDestFolder, 'dir')
        mkdir(subjectDestFolder);
    end
    for trial = 1:maxTrials
        fileName = sprintf('%s/Sub%02d/Data/Sub%02d_STAIR_%02d.mat',
baseFolder, subject, subject, trial);
        if exist(fileName, 'file')
            load(fileName, 'ACC_data', 'GYRO_data', 'EMG_data',
'TIME_data');

if length(ACC_data) ~= length(GYRO_data) || length(ACC_data) ~= length
(EMG_data) || length(GYRO_data) ~= length(EMG_data)
    %Resample
    ACC_data = [interp(ACC_data(:, 1), 4), interp(ACC_data(:, 2), 4),
interp(ACC_data(:, 3), 4)];
    GYRO_data = [interp(GYRO_data(:, 1), 4), interp(GYRO_data(:, 2), 4),
interp(GYRO_data(:, 3), 4)];
    % Synchronization of data lengths
    min_length = min([length(ACC_data), length(GYRO_data), length
(EMG_data)]);
    ACC_data = ACC_data(1:min_length, :);

    GYRO_data = GYRO_data(1:min_length, :);
    EMG_data = EMG_data(1:min_length, :);
    TIME_data = TIME_data(1:min_length, :);
end
GYRO_data_y = -GYRO_data(:, 2);
TIME_data = (0:0.005:(size(EMG_data(:, 1), 1) - 1) * 0.005)';
Fs = 200;
Fc = 10; % cut off frequency
Wn = Fc / (Fs / 2);

% Second order Butterworth filter
[b, a] = butter(2, Wn, 'low');
GYRO_data_y = filtfilt(b, a, GYRO_data_y);
% zero-crossings
ZP = findZeroCrossings(GYRO_data_y, 'positive');
ZN = findZeroCrossings(GYRO_data_y, 'negative');
```

```

% Initialize valid zero-crossings
ZP_valid = [];
ZN_valid = [];

% Threshold for signal segmentation
th=max(GYRO_data_y)/3;

for i=1:length(ZN)
    if ZN(1) < ZP(1) && i == 1
        % If the first point found is a ZN, skip the first iteration
        continue;
    elseif ZN(1) < ZP(1)
        % If ZN appears before ZP, use GYRO_data_y(ZP(i-1):ZN(i))
        segnale = GYRO_data_y(ZP(i-1):ZN(i));
    else
        % Otherwise, use GYRO_data_y(ZP(i):ZN(i))
        segnale = GYRO_data_y(ZP(i):ZN(i));
    end

    % Check if maximum value of segment exceeds the threshold
    if max(segnale) >= th
        if ZN(1) < ZP(1) && i > 1
            ZP_valid = [ZP_valid; ZP(i-1)];
            ZN_valid = [ZN_valid; ZN(i)];
        else
            ZP_valid = [ZP_valid; ZP(i)];
            ZN_valid = [ZN_valid; ZN(i)];
        end
    end
end

% Remove ZN and ZP if the interval between them is greater than 1.5 seconds
i = 1;
while i < length(ZN_valid)
    if TIME_data(ZP_valid(i+1)) - TIME_data(ZN_valid(i)) > 1.5
        % Remove ZN and ZP
        ZN_valid(i:i+1) = [];
        ZP_valid(i:i+1) = [];
        i = i + 1;
    else
        i = i + 1;
    end
end

% Remove the last ZP and ZN
ZP_valid(end-1:end) = [];
ZN_valid(end-1:end) = [];

% derivate of GYRO_data_y
GYRO_data_y_diff = diff(GYRO_data_y); %points where the signal changes
%first minimum after each ZN_valid point

```

```

IC = [];
for i = 1:length(ZN_valid)
    zn = ZN_valid(i);
    for j = zn:length(GYRO_data_y_diff)-1
        if GYRO_data_y_diff(j) < 0 && GYRO_data_y_diff(j + 1) >= 0
            IC = [IC; j+1];
            break;
        end
    end
end
end

FO = [];
for i = 1:length(ZP_valid)
    zp = ZP_valid(i);
    for j = zp-1:-1:1
        if GYRO_data_y_diff(j) < 0 && GYRO_data_y_diff(j + 1) >= 0
            FO = [FO; j];
            break;
        end
    end
end
end
if ~isempty(FO) && FO(1) < IC(1)
    FO = FO(FO > IC(1));
end

% label vector initialized with NaN
label = NaN(length(TIME_data), 2);
label(:, 1) = TIME_data;
label(:, 3) = 1;
for i = 1:length(IC)
    if i <= length(FO)
        % IC-FO
        start_index = find(TIME_data >= TIME_data(IC(i)), 1);
        end_index = find(TIME_data < TIME_data(FO(i)), 1, 'last');
        if ~isempty(start_index) && ~isempty(end_index) && (TIME_data(FO(
(i) - TIME_data(IC(i)) <= 2)
        label(start_index:end_index, 2) = 1;
    end
    %FO-IC
    if i < length(IC)
        start_index = find(TIME_data >= TIME_data(FO(i)), 1);
        end_index = find(TIME_data < TIME_data(IC(i+1)), 1,
'last');
        if ~isempty(start_index) && ~isempty(end_index) &&
(TIME_data(IC(i+1)) - TIME_data(FO(i)) <= 2)
            label(start_index:end_index, 2) = 2;
        end
    end
end
end
end

%table of cells to handle both numbers and strings

```

```

labelCell = num2cell(label);

% Replace NaNs with 'NaN' (string) in column 2
for i = 1:size(labelCell, 1)
    if isnan(labelCell{i, 2})
        labelCell{i, 2} = 'NaN';
    else
        labelCell{i, 2} = num2str(labelCell{i, 2});
    end
end

% Create a table
t = cell2table(labelCell, 'VariableNames', {'Time', 'Status', 'Group'});

% output file
outputFile = fullfile(subjectDestFolder, sprintf('Sub%02d_STAIR_%02d_Label.xls', subject, trial));

% Save in excel file
writetable(t, outputFile, 'FileType', 'spreadsheet');
else
    break;
end
end
end
end

```

### Codice Matlab per la funzione findZeroCrossing.m:

```

function ZC = findZeroCrossings(signal, slope)
    if strcmp(slope, 'positive')
        ZC = find(diff(signal > 0) == 1);
    else if strcmp(slope, 'negative')
        ZC = find(diff(signal > 0) == -1);
    end
end

% This function finds the indices of zero crossings in the given
signal
% The 'slope' parameter specifies the type of crossing:
% 'positive' for upward crossings and 'negative' for downward
crossings

```



## BIBLIOGRAFIA

- [1] M. Cody, W.-M. Sarah, W. Marcia, H. Juanita e K. Deborah, «Global prevalence of traumatic non-fatal limb amputation» *Prosthetics and Orthotics International*, 2021.
- [2] V. Arcovio, «Amputazioni, in Italia 15mila persone l'anno perdono un arto. Protesi hi-tech e il percorso per tornare a una vita normale» *Il Messaggero*, 2024.
- [3] H. T. T. Vu, F. Gomez, P. Cherelle, D. Lefeber, A. Now e B. Vanderborght, «ED-FNN: A New Deep Learning Algorithm to Detect Percentage of the Gait Cycle for Powered Prostheses» *sensors*, 2018.
- [4] O. A. Gonzales-Huisa, G. Oshiro, V. E. Abarca, J. G. Chavez-Echajaya e D. A. Elias, «EMG and IMU Data Fusion for Locomotion Mode Classification in Transtibial Amputees» *prosthesis*, 2023.
- [5] A. Cimolato, J. J. M. Driessen, L. S. Mattos, E. D. Momi, M. Laffranchi e L. D. Micheli, «EMG-driven control in lower limb prostheses: a topic-based systematic review» *Journal of NeuroEngineering and Rehabilitation*, 2022.
- [6] A. Mazumder, E. E. G. Hekman e R. Carloni, «Toward Controlling Transtibial Prostheses Using a Single Degree of Freedom Inertial Sensor System» *IEEEAccess*, 2024.
- [7] B. Ahkami, K. Ahmed, A. Thesleff, L. Hargrove e M. O. Catalan, «Electromyography-based control of lower limb prostheses: a systematic review» *IEEE transactions on medical robotics and bionics*, 2023.

- [8] W. Wei, F. Tan, H. Zhang, H. Mao e M. Fu, «Surface electromyogram, kinematic and kinetic dataset of lower limb walking for movement intent recognition» *Scientific data*, 2023.
- [9] W. Tao, T. Liu, R. Zheng e H. Feng, «Gait Analysis Using Wearable Sensors» *sensors*, 2012.
- [10] P. Fraccaro, L. Waish, J. Doyle e D. O'Sullivan, «Real-world Gyroscope-based gait event detection and gait feature extraction» in *The sixth International conference on eHealth, Telemedicine, and Social Medicine*, 2014.
- [11] A. Matos, T. Adão, L. Magalhães e E. Peres, «A myographic-based HCI solution proposal for upper limb amputees» *Conference on ENTERprise Information Systems / International Conference on ProjectMANagement / Conference on Health and Social Care Information Systems and Technologies*, 2016.
- [12] «Robot Domestici» 10 08 2024. [Online]. Available: <https://www.robot-domestici.it/joomla/home/myo-gesture-control-armband>.
- [13] L. Moreira, J. Figueiredo, P. Fonseca, J. P. Vilas-Boas e C. P. Santos, «Lower limb kinematic, kinetic, and EMG data from young healthy humans during walking at controlled speeds» *Scientific Data*, 2021.
- [14] J. Wojtusch e O. V. Stryk, «HUMOD - A versatile and open database for the investigation, modeling and simulation of human motion dynamics on actuation level» in *IEEE-RAS International Conference*, Seoul, 2015.

- [15] C. Morbidoni, A. Cucchiarelli, S. Fioretti e F. D. Nardo, «A Deep Learning Approach to EMG-Based Classification of Gait Phases during Level Ground Walking» *electronics*, 2019.
- [16] V. Agostini, M. Ghislieri, S. Rosati, G. Balestra e M. Knaflitz, «Surface Electromyography Applied to Gait Analysis: How to improve Its Impact in Clinics?» *frontiers in Neurology*, 2020.
- [17] B. Hu, E. Rouse e L. Hargrove, «Benchmark Datasets for Bilateral Lower-Limb Neuromechanical Signals from Wearable Sensors during Unassisted Locomotion in Able-Bodied Individuals» *frontiers in Robotics and AI*, 2018.
- [18] J. Camargo, A. Ramanathaz, W. Flanagan e A. Young, «A comprehensive, open-source dataset of lower limb biomechanics in multiple conditions of stairs, ramps, and level-ground ambulation and transitions» *Journal of biomechanics*, 2021.
- [19] L. Flynn, J. Geeroms, R. Jimenez-Fabian, S. Heins, B. Vanderborght, M. Munih, R. Molino Lova, N. Vitiello e D. Lefeber, «The challenges and achievements of experimental implementation of an active transfemoral prosthesis based on biological quasi-stiffness: The CYBERLEGS betaprosthesis» *Front. Neurorobot.*, 2018.
- [20] J. Taborri, E. Palermo, S. Rossi e P. Cappa, «Gait partitioning methods: A systematic review» *sensors*, 2016.
- [21] J. Taborri, S. Rossi, E. Palermo, F. Patanè e P. Cappa, «A novel HMM distributed classifier for the detection of gait phases by means of a wearable inertial sensor network» *sensors*, 2014.

- [22] N. Abaid, P. Cappa, E. Palermo, M. Petrarca e M. Porfiri, «Gait detection in children with and without» *Plos one*, 2013.
- [23] D. Gouwandaa e A. A. Gopalai, «A robust real-time gait event detection using wireless gyroscope and its application on normal and altered gaits» *Medical Engineering and Physics*, 2015.
- [24] H. F. Maqbool, M. A. Husman, M. I. Awad, A. Abouhoussein, P. Mehryar, N. Iqbal e A. Dehghani-Sanij, «Real-time gait event detection for lower limb amputees using a single wearable sensor» *IEEE*, 2016.
- [25] C. Senanayake e Senanayake, «S.M.N.A.Computational intelligent gait-phase detection system to identify pathological gait» *IEEE Trans. Inf. Technol. Biomed.*, 2010.
- [26] C. Joshi, U. Lahiri e N. Thakor, «Classification of gait phases from lower limb EMG: Application to exoskeleton orthosis» *In Proceedings of IEEE Point-of-Care Healthcare Technologies (PHT)*, 2013.
- [27] M. Alaqtash, H. Yu, R. Brower, A. Abdelgawad e T. Sarkodie-Gyan, «Application of wearable sensors for human gait analysis using fuzzy computational algorithm» *Eng. Appl. Artif. Intell*, 2011.
- [28] M. Salminen, J. Perttunene, J. Avela e A. Vehkaoja, «A novel method for accurate division of the gait cycle into seven phases using shank angular velocity» *Gait and Posture*, 2024.

- [29] A. Behboodi, S. C. K. Lee e N. Zahradka, «Seven phases of gait detected in real-time using shank attached gyroscope» in *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2015.
- [30] B. K. JK e A. K. G. S, «Assessing Smart Algorithms for Gait Phases Detection in Lower Limb Prosthesis: A Comprehensive Review» *Cornell University*, 2023.
- [31] B. K. JK e A. K. G. S, «Evaluating Intelligent Algorithms for Gait Phase Classification in Lower Limb Robotic Systems» *Robotics*, 2023.
- [32] J. Taborri, E. Palermo, S. Rossi e P. Cappa, «Gait Partitioning Methods: A Systematic Review» *sensors*, 2015.
- [33] H. T. T. Vu, D. Dong, H.-L. Cao, T. Verstraten, D. Lefeber, B. Vanderborght e J. Geeroms, «A Review of Gait Phase Detection Algorithms for Lower Limb Protheses» *sensors*, 2020.
- [34] W. Pirker e R. Katzenschlager, «Gait disorders in adults and the elderly» *The Central European Journal of Medicine*, 2016.
- [35] R. Luo, S. Sun, X. Zhang, Z. Tang e W. Wang, «A low-cost end-to-end sEMG-based gait sub-phase recognition system» *IEEE transactions on neural systems and rehabilitation engineering*, 2020.
- [36] H. Prasanth, M. Caban, U. Keller, G. Courtine, A. Ijspeert, H. Vallery e J. v. Zitzewitz, «Wearable Sensor-Based Real-Time Gait Detection: A Systematic Review» *sensors*, 2021.

- [37] E. Halilaj, A. Rajagopal, M. Fiterau, J. L. Hicks, T. J. Hastie e S. L. Delp, «Machine learning in human movement biomechanics: Best practices, common pitfalls, and new opportunities» *Journal of Biomechanics*, 2018.
- [38] P. Visconti, F. Gaetani, G. A. Zappatore e P. Primiceri, «Technical features and functionalities of Myo Armband: an overview on related literature and advanced applications of Myoelectric armbands mainly focused on arm prostheses» *International journal on smart sensing and intelligent systems*, 2018.
- [39] J. A. Mihy, M. Wagatsuma, S. M. Cain e J. F. Hafer, «Minimizing the Effect of IMU Misplacement With a Functional Orientation Method» *medRxiv*, 2022.
- [40] H. Ohashi, S. Ahmed, M. Al-Naser e T. Akiyama, «Augmenting Wearable Sensor Data with Physical Constraint for DNN-Based Human-Action Recognition» 2017.
- [41] A. Terranova, «Osteolab» 2022. [Online].
- [42] A. Anis, M. A. Irshad, S. M. Hamza, N. Naseer, H. Nazeer e Andrian, «EMG based control of transtibial prosthesis» *ReserarchGate*, 2020.
- [43] J. Wang, Y. Dai, T. Kang e X. Si, «Research on human motion recognition based on lower limb electromyography (EMG) signals» *IEEE 4th International conference on electronics technology*, 2021.
- [44] F. Gao, T. Tian, T. Yao e Q. Zhang, «Human gait recognition based on multiple feature combination and parameter optimization algorithms» *Computational Intelligence and Neuroscience*, 2021.

- [45] C. Tepe e M. C. Demir, «Detection and Classification of Muscle Activation in EMG Data Acquired by Myo Armband» *European Journal of Science and Technology*, 2020.
- [46] C. Tepe e M. Demir, «Real-Time Classification of EMG Myo Armband Data Using Support Vector Machine» *IRBM*, 2022.
- [47] A. Phinyomark e E. Scheme, «A Feature Extraction Issue for Myoelectric Control Based on Wearable EMG Sensors» *Institute of Biomedical Engineering university of New Brunswick*, 2018.
- [48] B. Ahkami, K. Ahmed, A. Thesleff, L. Hargrove e M. Ortiz-Catalan, «Electromyography-Based Control of Lower Limb Protheses: A Systematic Review» *IEEE transactions on medical robotics and bionics*, 2023.
- [49] A. Phinyomark, R. N. Khushaba e E. Scheme, «Feature extraction and selection for myoelectric control based on wearable EMG sensors» *sensors*, 2018.
- [50] Canva, [Online]. Available: <https://www.canva.com/>.
- [51] N. Olivarez-Giles, «Myo Motion-Sensing Armband: First Look at the Wearable's Finished Design» *The wall street journal*, 2014.
- [52] T. Dost, «ArtStation» [Online]. Available: <https://www.artstation.com/artwork/LAm4R>.
- [53] The MathWorks Inc., «Matlab» 2023b. [Online]. Available: <https://www.mathworks.com/products/matlab.html>.

- [54] G. Bastasa, J. J. Fleck, R. A. Peters e K. E. Zelik, «IMU-based gait analysis in lower limb prosthesis users: Comparison of step demarcation algorithms» *Gait & Posture*, 2018. [Online].
- [55] G. Prateek, P. Mazzoni, G. M. Earhart e A. Nehoral, «Gait Cycle Validation and Segmentation using Inertial Sensors» *IEEE Trans Biomed Eng*, 2020. [Online].
- [56] P. C. Formento, R. Acevedo, S. Ghoussayni e D. Ewins, «Gait Event Detection during Stair Walking Using a Rate Gyroscope» *sensors*, 2014.
- [57] A. Phinyomark, R. N. Khushaba e E. Scheme, «Feature Extraction and Selection for Myoelectric Control Based on Wearable EMG Sensor» *Sensors*, 2018.
- [58] J. Wu, H. Maurenbrecher, A. Schaer, B. Becsek, C. A. Easthope, G. Chatzipirpiridis, O. Ergeneman, S. Pané e B. J. Nelson, «Human gait-labeling uncertainty and a hybrid model for gait segmentation» *Frontiers in Neuroscience*, 2022.