POLITECNICO DI TORINO

Corso di Laurea Magistrale in Ingegneria Biomedica

Sviluppo, modellazione e integrazione di biofeedback visivi in un sistema di realtà virtuale per tele-riabilitazione dell'arto superiore

Relatore:
Prof. Ing. Andrea Cereatti
Prof. Ing. Danilo Pani
Dott. Ing. Giulia Sedda

Tesi di Laurea di:
Ferraro Pietro

Anno accademico 2025/2026

Alla memoria dei miei nonni: anche se non potete essere qui a condividere questo momento, vi porto nel mio cuore e vi dedico questo traguardo.

Abstract

Negli ultimi anni sono state sviluppate soluzioni di tele-riabilitazione innovative, basate su exergame e realtà virtuale (VR), capaci di combinare esercizio fisico, stimolazione cognitiva e sensoriale per fornire trattamenti efficaci a distanza. Un elemento chiave in questo contesto è l'integrazione di biofeedback, che guidino il paziente durante l'esecuzione dei movimenti e contribuiscano a migliorare l'efficacia del percorso riabilitativo. L'obiettivo del presente lavoro di tesi è stato quello di progettare e sviluppare in Unity 3D diversi biofeedback primariamente visivi, basati su dati sia cinematici che di attivazione muscolare (sEMG), e di integrarli in un'applicazione di tele-riabilitazione in VR per l'arto superiore, sviluppata per il progetto MYRTUS. Per poter partecipare al training, ciascun paziente deve indossare un caschetto VR per la visualizzazione degli stimoli visivi e impugnare i controller per il tracciamento dei movimenti delle mani nello spazio 3D. L'exergame proposto si basa su un esercizio di canottaggio sincronizzato, in cui più pazienti devono trovare una strategia motoria comune per far muovere la barca in un fiume. Il movimento che devono compiere è un reaching planare, mappato nell'ambiente VR con il gesto della remata. I biofeedback forniti all'utente durante l'allenamento sono stati progettati per fornire informazioni in tempo reale e personalizzate, relative alla completezza e correttezza del movimento eseguito. Tra i biofeedback sviluppati, uno di essi si basa sull'iperattivazione anomala del muscolo deltoide laterale, che in pazienti post-ictus è dovuta alla spasticità muscolare e a movimenti compensatori di abduzione di spalla e intra-rotazione del braccio. Nell'ottica di generare un biofeedback soggetto-specifico in grado di rilevare tali compensazioni e fornire contestualmente indicazioni sulla sovra-attivazione muscolare, è stato sviluppato un modello di regressione non lineare basato sul segnale sEMG del deltoide laterale, a partire da movimenti di abduzione e adduzione. Per la creazione e validazione del modello è stata condotta un'analisi sperimentale su 15 partecipanti sani, durante l'esecuzione di movimenti di reaching planare con o senza movimenti di abduzione di spalla. L'attività elettromiografica del deltoide laterale e gli angoli articolari di spalla sono stati acquisiti mediante un sistema inerziale indossabile e sono stati successivamente elaborati in MATLAB. I risultati preliminari mostrano che il modello sviluppato è in grado di discriminare la presenza di abduzioni anomale con differenze statisticamente significative tra le diverse condizioni sperimentali. Il confronto tra i range angolari stimati con quelli rilevati dai sensori inerziali ha confermato l'efficacia del modello nell'identificare correttamente le abduzioni anomale. Tuttavia, l'elevata variabilità inter-soggetto suggerisce la necessità di estendere lo studio ad un campione sperimentale maggiore. La validazione di tutti i biofeedback su partecipanti post-stroke rappresenta il principale sviluppo futuro, volto a verificare la rilevanza delle metodologie e degli strumenti sviluppati in contesti clinici reali.

Questo lavoro è stato supportato dal progetto MYRTUS, finanziato dall'Unione Europea, grant No. 101135183 (le opinioni espresse sono esclusivamente quelle dell'autore e non riflettono necessariamente quelle dell'Unione Europea; né l'Unione Europea né l'ente finanziatore possono essere ritenuti responsabili per esse).

Sommario

Lista delle figure ix
Lista delle tabelle x
Introduzione 1
La teleriabilitazione per il recupero neuromotorio 1
Obiettivi 2
Struttura e organizzazione della tesi 3
Stato dell'arte 4
Biofeedback 4
Caratteristiche e criteri di progettazione 4
Aree di applicazione 6
Teleriabilitazione degli arti superiori 7
Serious game e approcci collaborativi 7
Teleriabilitazione basata sulla Realtà Virtuale 10
Realtà virtuale non immersiva 10
Realtà virtuale immersiva 15
Movimenti compensatori durante il reaching 20
Introduzione 20
Biomeccanica del reaching e analisi delle compensazioni dei pazienti post-ictus 20
Rilevazione, valutazione e correzione dei movimenti compensatori 25
Materiali e metodi 29
Architettura del sistema 29
Integrazione della strumentazione in ambiente Unity 3D 30
Caratteristiche dei sensori 31
Implementazione software 34
Procedura di calibrazione 43
Panoramica della procedura 44
Descrizione dei Game Object principali 45
Interfacce utente e logica di interazione 49
Task motorio e logica di controllo dei remi 58
Mapping da movimento lineare a movimento circolare 58
Logica di controllo del remo 61

Progettazione e implementazione dei Biofeedback 63
Biofeedback cinematici 64
Biofeedback EMG 70
Modellazione e valutazione preliminare del BF basato su EMG 72
Partecipanti e apparato sperimentale 72
Protocollo sperimentale 73
Valutazione preliminare dell'attività muscolare durante l'abdo-adduzione 77
Creazione del modello 80
Applicazione del modello 83
Analisi dei dati 87
Risultati e discussione 88
Selezione delle ripetizioni valide e segmentazione 88
Calibrazione del modello: scelta della funzione di fitting 94
Risultati del modello 98
Conclusioni 106
Ringraziamenti 108
Bibliografia 109

Lista delle figure

Figura 1: Esempi di ADL presenti nella piattaforma SaeboVR. *11*
Figura 2: Il sistema AR Rehab per l'allenamento virtuale. *12*
Figura 3: Il kit DoMo del sistema DoMoMEA, ... *13*
Figura 4: Esempio di sistemi per il tracciamento dei movimenti della mano. *14*
Figura 5: Scenario applicativo dell'ambiente VR immersivo sviluppato da [60] *16*
Figura 6:Architettura del sistema RecoveryFun. *17*
Figura 7: Esercizi proposti dalla piattaforma sviluppata da Shih et al. (2024) [71]. *19*
Figura 8: Profilo di velocità della mano in un paziente post ictus per il braccio più colpito (MA) e quello meno colpito (LA) durante un compito di reaching. *21*
Figura 9: Strategie compensatorie durante l'esercizio di reaching. *22*
Figura 10: Contributi cinematici durante il forward e hand-to-mouth reaching nei pazienti sani e patologici *23*
Figura 11: Architettura del sistema proposto: *29*
Figura 12: Visuale locale (lato client) degli utenti nella barca virtuale. *30*
Figura 13: Visore Meta Quest 3 e controller Meta Quest Touch Plus *31*
Figura 14: Sistema Due+Pro. *33*
Figura 15: Schema delle connessioni nella modalità PC-Link nel caso di utilizzo della SyncStationBipolar (sistema Due+pro) *36*
Figura 16: Schema delle connessioni nella modalità stand-alone nel caso di utilizzo della sonda singola (sistema due+Lite) *37*
Figura 17: Game Object presenti nella Scena di calibrazione *45*
Figura 18: Interfaccia SetLanguage e relativi GO. *49*
Figura 19: Interfaccia WelcomeView e relativi GO. *50*
Figura 20: Interfaccia ConnectingView e relativi GO. *51*
Figura 21: Interfaccia CalibrationView e relativi GO. *51*
Figura 22: Interfacce utente per la calibrazione del ROM. *52*
Figura 23: CalibrationView nelle varie fasi della calibrazione del ROM . A) Countdown. B) Posizione al petto. C) Posizione iniziale. D) Posizione finale. *53*
Figura 24: Interfacce per l'acquisizione MVC. A) CalibrationView durante la fase di Calibrazione EMG. B) Pop-up Calibrazione EMG. *54*
Figura 25:CalibrationView nelle varie fasi dell'acquisizione dell'MVC: A) Countdown. B) Acquisizione del rumore di fondo (muscolo a riposo). C)Esecuzione della contrazione massimale. D) Fase di recupero. *55*
Figura 26: TestView e relativi GO. *56*
Figura 27: Rappresentazione della traiettoria circolare descritta dal mapping angolare. *60*
Figura 28: Posizionamento del Canvas contenete gli elementi grafici relativi ai biofeedback rispetto all'avatar virtuale. *63*
Figura 29: Rappresentazione del biofeedback associato al ROM nei casi di movimento completo o incompleto *65*
Figura 30: Rappresentazione grafica del biofeedback sulla velocità *67*
Figura 31: Sequenza di frame che mostrano l'effetto della coroutine per sollevare il remo dall'acqua quando vengono rilevati dei movimenti compensatori. *70*
Figura 32: Apparato e set-up sperimentale. *73*
Figura 33: T-pose assunta dal soggetto per la calibrazione del software di ricostruzione articolare, acquisizione della MVC, esercizio di abdo-adduzione. *74*
Figura 34: I tre esercizi di reaching planare (point-to-point) per la validazione del modello. . *75*
Figura 35: Andamento della relazione media dell'inviluppo EMG normalizzato in funzione delle variazioni dell'angolo di abdo-adduzione calcolata su tutto il dataset sperimentale.. ..... *79*
Figura 36: Effetto della velocità sul muscolo deltoide laterale nelle condizioni senza carico , con 25% NMW e 50% NMW [127] *85*

Figura 37: Rappresentazione grafica della funzione di warping sigmoidale in funzione della percentuale del range del modello *86*
Figura 38. Visualizzazione dei criteri di inclusione/esclusione basati sulle derivate cinematiche su un soggetto con tutte le ripetizioni valide. *89*
Figura 39: Visualizzazione dei criteri di inclusione/esclusione basati sulle derivate cinematiche su un soggettocon esclusione parziale singola. *89*
Figura 40: Visualizzazione dei criteri di inclusione/esclusione basati sulle derivate cinematiche su un soggetto con nessuna ripetizione valida.. ..... *90*
Figura 41 Esempio di identificazione degli istanti di inizio e fine movimento relativi. *91*
Figura 42: Esempio di identificazione degli istanti di inizio e fine movimento *91*
Figura 43: Quantificazione del segnale escluso dalla procedura di rimozione delle non-monotonie. *92*
Figura 44: Distribuzione dei valori di Adjust-R^2 di tutte le funzioni di fitting testate, calcolate sull'intero dataset di ripetizioni valide *95*
Figura 45: Distribuzione dei valori di RMSE di tutte le funzioni di fitting testate, calcolato sull'intero dataset di ripetizioni valide *95*
Figura 46: Confronto tra le funzioni di fitting migliori *97*
Figura 47: Confronto tra le funzioni di fitting migliori. *97*
Figura 48: Esempio di esclusione per scarsa qualità del segnale cinematico *98*
Figura 49 Esempio di esclusione per incongruenza cinematica rispetto al protocollo sperimentale. *98*
Figura 50: Variazione angolare di abduzione ricostruita dal modello durante i task di reaching in appoggio (task 1), senza appoggio (task 2) e con simulazione di movimenti compensatori (task 3) best case *99*
Figura 51: Variazione angolare di abduzione ricostruita dal modello durante i task di reaching in appoggio (task 1), senza appoggio (task 2) e con simulazione di movimenti compensatori (task 3) worst case. *99*
Figura 52: Variazione angolare di abduzione ricostruita dal modello durante i task di reaching in appoggio (task 1), senza appoggio (task 2) e con simulazione di movimenti compensatori (task 3) suboptimal case *100*
Figura 53: Confronto tra le distribuzioni di Range Angolari dei dati IMU (box blu) e ricostruiti (box arancioni) per i tre Task *101*
Figura 54: Analisi degli errori dei range angolari calcolati come differenza tra il dato reale (IMU) e quello ricostruito. *102*
Figura 55: Istogramma di densità bidimensionale dei dati IMU e ricostruiti. *104*

Lista delle tabelle

Tabella 1: Funzioni di fitting analizzate per la creazione del modello *81*
Tabella 2: Risultati dell'algoritmo di esclusione sui movimenti di abdo-adduzione utilizzati per il training del modello. La tabella illustra la classificazione in prove "valide" e "non valide" per ciascuna delle tre ripetizioni eseguite dai 15 soggetti sperimentali (Subj 01 - 15). *88*
Tabella 3: Confronto delle performance di fitting per le diverse funzioni matematiche testate. *94*
Tabella 4: Valori medi e deviazione standard del Range Angolare dei dati ricostruiti dai sensori inerziali (IMU) e prodotti dal modello per ogni ripetizione del dataset e relativo Errore di stima. *102*

Introduzione

Questo lavoro è stato svolto in collaborazione con l'Università degli Studi di Cagliari, presso il laboratorio MeDSP (*Medical Devices and Signal Processing*) del Prof. Danilo Pani e supportato dal progetto MYRTUS, finanziato dall'Unione Europea, con il grant No. 101135183 (le opinioni espresse sono esclusivamente quelle dell'autore/degli autori e non riflettono necessariamente quelle dell'Unione Europea; né l'Unione Europea né l'ente finanziatore possono essere ritenuti responsabili per esse).

La teleriabilitazione per il recupero neuromotorio

Alla fine degli anni Novanta, con l'introduzione e la progressiva diffusione di Internet, l'Organizzazione Mondiale della Sanità, nel documento *"A Health Telematics Policy in Support of WHO's Health-for-All Strategy"* (WHO, 1998) [1], riconosce nelle tecnologie dell'informazione e della comunicazione uno strumento chiave per migliorare accesso, qualità ed equità dei servizi sanitari, soprattutto nelle aree rurali e svantaggiate, definendo la telemedicina come: "l'erogazione di servizi sanitari a distanza, quando la distanza rappresenta un fattore critico, da parte dei professionisti della salute che utilizzano le tecnologie dell'informazione e comunicazione per scambiare informazioni clinicamente valide ai fini della diagnosi, del trattamento e della prevenzione delle malattie, nonché per la ricerca, la valutazione e la formazione continua degli operatori, con l'obiettivo di promuovere la salute degli individui e delle comunità" [1].

La teleriabilitazione si inserisce nel più ampio panorama della telemedicina come risposta al crescente fabbisogno globale di interventi riabilitativi [2] con l'obiettivo di fornire modelli assistenziali più sostenibili, capaci di trasferire parte della presa in carico dall'ospedale al domicilio [3]. Dal punto di vista clinico, la teleriabilitazione comprende una serie di servizi di riabilitazione e abilitazione che includono valutazione, monitoraggio, prevenzione, intervento, supervisione, istruzione, consulenza e assistenza [4].Il primo decennio del XXI secolo ha visto un'impennata dei programmi di telemedicina e teleriabilitazione, attribuibile sia ai progressi tecnologici che alla maggiore consapevolezza dei suoi potenziali benefici tra cui:

- Aumentare l'accessibilità e la continuità della riabilitazione nei soggetti con disabilità, in particolare in popolazioni vulnerabili e geograficamente svantaggiate [5].
- Migliorare la qualità della vita: condurre alcuni aspetti della riabilitazione all'interno dell'ambiente naturale, casa o lavoro, aumenta i risultati funzionali, migliora la generalizzazione dei progressi (*transfer of learning*) e la soddisfazione del paziente.[6]
- Ridurre le spese mediche, la necessità di spostamento e accogliere un numero maggiore di pazienti. Inoltre la teleriabilitazione risulta più flessibile rispetto alla riabilitazione tradizionale, agevolando la pianificazione delle sedute in base alle disponibilità del paziente e del terapista [7].
- Efficacia del trattamento comparabile con la terapia convenzionale [8], [9], in condizioni ben definite in termini di tipologia e severità della patologia [10].

La teleriabilitazione permette inoltre di ampliare gli orizzonti della riabilitazione classica mediante l'utilizzo di tecnologie innovative. In particolare la realtà virtuale (VR) offre un'esperienza immersiva che aumenta il coinvolgimento e la motivazione del paziente, migliora l'aderenza al trattamento [11], nei pazienti per i quali la VR è applicabile. Questa tecnologia viene spesso associata ai *serious game*, ovvero giochi digitali progettati con finalità educative o terapeutico, che vengono definiti *exergame* quando l'interazione con il sistema richiede l'esecuzione di movimenti corporei reali, coniugando l'esercizio fisico alla componente ludica [12], [13].

Accanto a questi benefici, emergono diversi limiti e criticità che ostacolano l'utilizzo della telemedicina: l'assenza del contatto fisico per le terapie manuali, la difficoltà a sostituire completamente l'esame obiettivo in presenza, il rischio di indebolire la relazione medico–paziente, il persistere di disuguaglianze di accesso legate al *digital divide*, applicabilità limitata in pazienti con deficit cognitivi significativi, nonché la necessità di ingenti investimenti in infrastrutture, personale e formazione. Permangono inoltre nodi aperti sul piano normativo, regolatorio ed etico (licenze professionali, responsabilità, privacy e sicurezza dei dati) [14], [15], [16], [17].

Con l'arrivo della pandemia di COVID-19, la necessità di garantire continuità assistenziale limitando gli accessi in presenza ha portato ad una rapida evoluzione dei servizi a distanza: molti Paesi hanno allentato le restrizioni legali, ampliato i rimborsi e adottato la telemedicina in numerosi ambiti, segnando un'ulteriore svolta nello sviluppo della sanità digitale [18].

Un aspetto fondamentale della teleriabilitazione neuromotoria, specialmente se svolta in assenza della supervisione diretta del terapista, è guidare il soggetto nella corretta esecuzione dell'esercizio riabilitativo. Tale obiettivo viene raggiunto mediante l'integrazione di sistemi capaci di analizzare in tempo reale i segnali fisiologici (come l'attività muscolare) e i parametri biomeccanici dell'utente fornendogli un'informazione di ritorno (biofeedback) intuitiva sulla qualità dell'esecuzione motoria. Queste tecnologie non solo supportano il paziente durante la sessione, ma permettono anche il monitoraggio dei progressi [19].

Obiettivi

Il presente lavoro di tesi si inserisce nell'ambito del sistema di teleriabilitazione, sviluppato dall'Università degli Studi di Cagliari come *Use Case Healthcare* dell'architettura MYRTUS. Lo *Use Case Healthcare* propone un *exergame* multiutente in realtà virtuale (VR), sviluppato nella piattaforma Unity, per la riabilitazione neuromotoria dell'arto superiore. Il protocollo riabilitativo è basato su un esercizio collaborativo di canottaggio, nel quale ogni soggetto controlla un solo remo virtuale attraverso un gesto di *reaching* planare. L'*exergame* consiste nell'esplorazione di un fiume-labirinto dove ogni partecipante, nonostante una visuale limitata del percorso e degli altri rematori, deve sincronizzarsi con l'equipaggio per completare il tragitto. In questo contesto, dinamiche fondamentali come la virata e la gestione della velocità emergono esclusivamente dallo sforzo collettivo senza alcuna interazione verbale o coordinazione supervisionata. Il sistema promuove così la partecipazione attiva di tutti i coinvolti, rendendo il contributo di ogni singolo utente indispensabile per il raggiungimento dell'obiettivo comune.

Il contributo di questo studio allo *Use Case Healthcare*, nonché l'obiettivo principale della tesi, risiede nella progettazione, implementazione e integrazione di un sistema di *biofeedback* visivi. Lo scopo dei biofeedback è informare il partecipante sulla propria prestazione motoria durante l'esecuzione del compito riabilitativo, anche in assenza del fisioterapista. La correttezza del movimento è valutata sia in termini cinematici (estensione del movimento e velocità), sia in termini di assenza di sovra-attivazioni muscolari da parte del deltoide laterale, che nei soggetti patologici (es. post-ictus) sono riconducibili a movimenti compensatori di abduzione della spalla (elevazione e intra-rotazione del gomito).

Per il conseguimento di tale scopo sono stati definiti i seguenti obiettivi specifici:

1. Integrazione del setup sperimentale su Unity, ovvero l'*headset* VR (Meta Quest 3) e i sensori EMG (Due+ di OT Bioelettronica) e implementazione di una *pipeline* software per l'elaborazione dei segnali in tempo reale e il salvataggio dei dati.
2. Implementazione di un algoritmo di *mapping* del movimento lineare del *reaching* planare in una traiettoria circolare per permettere il controllo del remato nell'ambiente virtuale.
3. Progettazione e implementazione su Unity di una procedura di calibrazione per la stima del *Range of Motion* (ROM) e la Massima Contrazione Volontaria (MVC). Il ROM viene stimato per standardizzare la sessione in base alla capacità motorie del soggetto, mentre l'MVC per la normalizzazione dell'EMG e la definizione delle soglie di attivazione muscolare.
4. Sviluppo e validazione preliminare di un modello di regressione soggetto-specifico basato su segnale elettromiografico per la rilevazione in tempo reale dei movimenti compensatori di abduzione.

Struttura e organizzazione della tesi

Il presente lavoro di tesi si articola in cinque capitoli compreso il presente. Nel secondo capitolo verrà fornita una panoramica dello stato dell'arte relativo alla progettazione e applicazione dei biofeedback in riabilitazione e teleriabilitazione. Inoltre, verrà approfondita la teleriabilitazione degli arti superiori e i sistemi che utilizzano *serious games* e realtà virtuale. Infine, verranno descritte le strategie di compenso motorio dei soggetti patologici durante il movimento di *reaching* e le tecniche utilizzate per la loro rilevazione.

Nel terzo capitolo saranno illustrate le metodologie impiegate per il raggiungimento degli obiettivi prefissati. La sezione include la descrizione dell'apparato e del protocollo sperimentale, unitamente ai dettagli tecnici riguardanti le implementazioni software su Unity e le scelte progettuali adottate durante lo sviluppo.

Il quarto capitolo sarà dedicato alla presentazione e alla discussione dei risultati ottenuti. Infine, nell'ultimo capitolo saranno presenti le conclusioni e gli sviluppi futuri sul lavoro svolto

Stato dell'arte

Biofeedback

Nel contesto riabilitativo, il biofeedback può essere definito come una tecnica che fornisce, in tempo reale, informazioni oggettive su processi fisiologici o biomeccanici che normalmente non sarebbero direttamente e facilmente controllabili dal paziente. Si tratta, in altri termini, di un feedback aumentato o estrinseco, proveniente da fonti esterne come un terapista o un sistema digitale, che integra le informazioni naturalmente fornite dal corpo umano attraverso il sistema sensoriale (feedback intrinseco) [20]. Questo aspetto assume un'ulteriore rilevanza quando il sistema di feedback intrinseco risulta compromesso a seguito di un danno cerebrale, ad esempio in pazienti post-ictus [21]. L'impiego del biofeedback in riabilitazione è ampiamente documentato con lo scopo di facilitare il recupero, migliorare l'apprendimento motorio, l'aderenza ai programmi riabilitativi, la consapevolezza della tecnica di esecuzione e aumentare sia la *performance* (prestazione) motoria che funzionale, generando al contempo misure oggettive utili per fissare obiettivi riabilitativi e monitorare i progressi nel tempo [20], [21].

Caratteristiche e criteri di progettazione

Brennan et al. (2020) [22] e Sigrist et al. (2013) [23] offrono un'analisi dettagliata sulle caratteristiche dei biofeedback e il loro utilizzo durante le varie fasi della riabilitazione. La classificazione si basa su quattro componenti fondamentali del feedback: modalità, contenuto, frequenza e tempistica. Questa suddivisione permette di descrivere con precisione qualsiasi tipologia di feedback e costituisce un riferimento essenziale per la progettazione di sistemi digitali [22].

La modalità definisce attraverso quale canale sensoriale il feedback viene trasmesso all'utente; le principali modalità utilizzate in ambito riabilitativo e di controllo motorio sono:

- Visiva: è la modalità più comunemente utilizzata. Il feedback visivo può essere diretto, in cui la variabile misura viene mostrata numericamente o indiretto, in cui tale informazione viene trasformata ad esempio animando un avatar o una barra. I feedback visivi possono aumentare il coinvolgimento, distrarre dal compito, comunicare concetti quali l'estensione del movimento o la *performance* in modo chiaro e oggettivo [22].

- Uditiva: questa modalità è particolarmente indicata per trasmettere informazioni temporali (ritmo, periodicità) o per ridurre il carico visivo quando il paziente deve già monitorare visivamente l'ambiente o il proprio corpo. Il feedback audio può variare da semplici allarmi (es. un suono quando si supera una soglia) alla sonificazione del movimento, dove i parametri continui del suono (tono e volume) mappano l'esecuzione motoria in tempo reale [22], [23].

- Tattile: nella quale vengono stimolati i recettori della pelle attraverso vibrazioni (feedback vibrotattile), pressione o texture superficiali di diverso tipo (soffice, rugoso, duro, scivoloso, etc) [23].

- Cinestetico: che prevede l'utilizzo di forze applicate all'effettore terminale (es. la mano) per guidare il paziente durante il movimento. Le forze possono essere applicate direttamente dal

terapista o da dispositivi specifici come ad esempio i robot riabilitativi [22]. Questa tipologia di feedback è molto utilizzata per percepire direttamente le conseguenze di una certa azione, caratteristica che lo rende particolarmente adatto a supportare l'apprendimento basato sull'errore e sull'esplorazione motoria [23]. Questa tipologia di feedback viene definita anche "aptica"[23]

- Multimodale: l'integrazione di più modalità sensoriali (es. audiovisiva) per mediare l'apprendimento è spesso superiore all'uso unimodale, poiché distribuisce le informazioni su diversi canali sensoriali riducendo il sovraccarico cognitivo [22], [23].

La decisione su quando e come fornire il feedback influisce direttamente sui meccanismi di correzione dell'errore. La tempistica di presentazione del feedback può essere simultanea, cioè il feedback viene erogato durante l'esecuzione del movimento, o alla fine del movimento o della sessione (terminale). La frequenza con cui viene fornito il feedback può essere costante, viene sempre fornito un feedback durante l'esecuzione, o decrescente, ovvero decresce col tempo (*faded feedback*), tipicamente in risposta ai miglioramenti della *performance* del soggetto [22]. Un approccio alternativo è il feedback a frequenza ridotta, ovvero viene fornito solo se si supera una certa soglia di errore (*bandwidth feedback*), oppure la frequenza viene adattata in tempo reale in base alle *performance* del paziente (*performance-based adaptation)* o infine è il soggetto stesso che decide quando ricevere il feedback (s*elf-controlled feedback*) [23]. Fornire immediatamente il feedback e in maniera costante è fondamentale nelle fasi iniziali dell'apprendimento o per compiti complessi, in quanto guida il paziente verso l'esecuzione corretta e previene il consolidamento di schemi motori errati [23]. Tuttavia, un uso prolungato può generare l'effetto di *guidance hypothesis,* in quanto il paziente si affida al feedback esterno trascurando i processi interni di rilevamento dell'errore, con un calo della *performance* non appena il feedback viene rimosso [23]. L'approccio terminale, sebbene meno efficace nel correggere l'errore "in diretta", è superiore per la memoria a lungo termine e il trasferimento dell'apprendimento, poiché costringe il paziente a riflettere sulla propria esecuzione e a pianificare il movimento successivo basandosi sulla memoria [23]. Parallelamente la frequenza dovrebbe diminuire con l'aumentare delle abilità del soggetto, nelle fasi avanzate servono infatti più prove senza feedback per consolidare l'apprendimento nel lungo periodo [23].

Il contenuto del feedback si divide in: conoscenza dei risultati (*Knowledge of Results*- KR) e conoscenza delle prestazioni (K*nowledge of Performance* - KP). Nel contesto della riabilitazione, il KR è l'informazione relativa al risultato del movimento, ad esempio il numero di ripetizioni raggiunte, mentre il KP è l'informazione relativa all'esecuzione del movimento. Il contenuto del feedback KP può essere descrittivo (una descrizione degli errori) o prescrittivo (una raccomandazione per la correzione degli errori) [22]. KR e KP svolgono funzioni diverse: il primo è motivante ed esplicativo, il secondo è essenziale per guidare la tecnica [22]. Sigrist et al. (2013) aggiungono che nelle fasi iniziali o nei compiti complessi il KP prescrittivo risulta fondamentale, poiché l'utente non è ancora in grado di interpretare autonomamente l'errore [23]. Ne deriva che nelle prime fasi va privilegiato il KP prescrittivo, integrato con un KR per rinforzo, mentre nelle fasi avanzate il KP descrittivo e il KR diventano più efficaci per sviluppare autonomia e auto-correzione.

L'integrazione tra le evidenze teoriche di Sigrist et al. (2013) e l'analisi applicativa di Brennan et al. (2020) mostra che la progettazione del feedback richiede un approccio sistemico e dinamico. Modalità, contenuto, frequenza e tempistica devono essere selezionati sulla base della complessità del compito, della fase di apprendimento, delle capacità sensoriali e cognitive dell'utente. Solo una progettazione consapevole e adattiva può tradursi in sistemi di feedback realmente efficaci nel contesto riabilitativo [22] [23].

Aree di applicazione

La versatilità dei sistemi di biofeedback ne ha permesso l'adozione in molteplici contesti clinici, spaziando dalla riabilitazione neurologica alla gestione di patologie sistemiche e muscoloscheletriche.

In ambito neurologico, il biofeedback, in particolare quello elettromiografico (EMG-BF), rappresenta uno strumento consolidato per la rieducazione neuromuscolare. Il biofeedback EMG consiste nel prelevare tramite elettrodi i segnali mioelettrici del muscolo, per rilevare un cambiamento nell'attività muscolare, per poi trasmettere questa informazione all'utente solitamente tramite un segnale visivo o uditivo [20]. Il biofeedback EMG può essere utilizzato sia per aumentare l'attività dei muscoli deboli o paretici, sia per facilitare una riduzione del tono muscolare spastico e trattare condizioni complesse come la sindrome spalla-mano [20], [24], [25]. Inoltre, trova applicazione nelle patologie del sistema muscoloscheletrico legate al dolore cronico, come la lombalgia e la cervicalgia, nelle disfunzioni del pavimento pelvico, disfagia, recupero dopo lesioni dei nervi periferici e controllo protesico [19], [20], [26]. I biofeedback sono applicati con successo anche nella riabilitazione cardiovascolare e respiratoria. Il monitoraggio della variabilità della frequenza cardiaca è utilizzato per la gestione dell'ipertensione, dell'insufficienza cardiaca e di condizioni legate allo stress e all'ansia. In ambito respiratorio, il feedback visivo o uditivo guida i pazienti nell'apprendimento di tecniche di respirazione diaframmatica per gestire la dispnea derivante da patologie come la broncopneumopatia cronica ostruttiva (BPCO) o l'asma [20].

Infine, nell'analisi del movimento possono essere forniti vari biofeedback in funzione dei parametri misurati tra cui [27]:

- Variabili cinematiche: forniscono informazioni sui movimenti articolari o segmentari (ad esempio angoli o *Range of Motion*- ROM) e comprendono parametri spaziotemporali del movimento, come velocità, durata o simmetria del gesto, che aiutano a migliorare la fluidità e la coordinazione.
- Variabili cinetiche: riguardano le forze generate durante il movimento, come la pressione esercitata o la spinta, e sono utili per correggere schemi motori alterati e rieducare l'equilibrio e la coordinazione motoria [20].

I biofeedback svolgono un ruolo cruciale nella teleriabilitazione: guidano il paziente durante le sessioni, permettendo il monitoraggio dei progressi e favoriscono un supporto personalizzato con conseguente aumento dell'aderenza al trattamento [19].

Teleriabilitazione degli arti superiori

La teleriabilitazione trova applicazione in un ampio spettro di condizioni cliniche, dalla neuroriabilitazione alla gestione di patologie muscoloscheletriche, cardiologiche e oncologiche [3], [5], [28].

In questo vasto scenario, un ruolo di primaria importanza è rivestito dalla disabilità degli arti superiori. Essa rappresenta un problema di salute di grande rilevanza globale, sia per gli elevati costi economici associati alle cure, sia per il marcato impatto negativo sulla qualità di vita degli individui [29].

Tale condizione deriva da diverse patologie tra cui: l'ictus, lesioni cerebrali acquisite, lesioni midollari, malattia di Parkinson, fratture traumatiche, patologie muscoloscheletriche, sclerodermia sistemica, artrite reumatoide e paralisi cerebrale emiplegica [30]. Tra queste, l'ictus suscita grande interesse nella ricerca scientifica in quanto rappresenta la seconda causa di morte nel mondo e si prevede che colpisca una persona su quattro di età superiore ai 25 anni. Rappresenta, inoltre, la terza causa principale se si considerano congiuntamente mortalità e disabilità. Le disabilità motorie degli arti superiori sono molto comuni tra i sopravvissuti all'ictus, colpendo circa il 90% dei pazienti acuti e persistendo fino al 60% dei casi cronici [31].

L'apprendimento motorio rappresenta uno dei principali riferimenti teorici per la riabilitazione neurologica. Secondo Levin e Demers (2021) [32], il recupero delle funzioni dell'arto superiore dopo una lesione del sistema nervoso centrale non dipende semplicemente dalla ripetizione automatica degli esercizi, ma richiede una pratica strutturata che consideri l'interazione tra individuo, compito e ambiente. Un elemento centrale è l'intensità dell'allenamento: per indurre cambiamenti stabili sono necessarie numerose ripetizioni dei compiti motori, distribuite in sessioni sufficientemente prolungate nel tempo.

Parallelamente, i pazienti con compromissione degli arti superiori tendono a trascorrere periodi di degenza sempre più brevi e vengono dimessi precocemente, principalmente a causa dei vincoli temporali e delle considerazioni economiche a cui i sistemi sanitari sono sottoposti. Tuttavia, molti di essi necessitano ancora di interventi riabilitativi prolungati per conseguire un recupero funzionale ottimale. Tali servizi risultano spesso onerosi e richiedono, in numerosi casi, accessi ripetuti presso strutture specialistiche, non sempre facilmente raggiungibili, con conseguente aumento dei costi indiretti e del carico organizzativo per pazienti e caregiver [33]

In questo contesto, la teleriabilitazione si configura come una modalità assistenziale promettente per favorire la continuità del trattamento, incrementare l'intensità della pratica motoria e ridurre i costi complessivi, consentendo al paziente di svolgere parte del percorso riabilitativo nel proprio ambiente domestico.

Serious game e approcci collaborativi

I paradigmi di teleriabilitazione possono essere sincroni, asincroni o misti. Gli interventi sincroni prevedono che i pazienti vengano seguiti individualmente o in gruppo da terapisti in tempo reale tramite videoconferenza, mentre quelli asincroni consistono in sessioni eseguite in modo autonomo dal paziente [34], e i cui dati sono inviati in differita al terapista per la

valutazione. La teleriabilitazione sincrona può presentare alcune limitazioni, in particolare in termini di scalabilità del trattamento e vincoli di programmazione, fattori che hanno alimentato l'interesse verso modalità asincrone [19].

Inizialmente, la teleriabilitazione asincrona consisteva nella semplice implementazione di piattaforme web che consentivano ai terapisti di prescrivere programmi di esercizi domiciliari personalizzati, corredati da video dimostrativi, istruzioni strutturate e strumenti di comunicazione diretta. Il paziente esegue gli esercizi a casa seguendo le indicazioni ricevute, mentre il professionista può monitorare l'andamento e, almeno in parte, la qualità dell'esecuzione attraverso report automatici o sessioni di teleconsulto [34].

Tuttavia, i programmi di esercizio terapeutico, svolti regolarmente a domicilio o in contesti non supervisionati, possono essere facilmente compromessi da numerosi fattori psicologici, medici e ambientali tra loro interconnessi, che rappresentano delle barriere al pieno coinvolgimento nella riabilitazione [35]. Parallelamente, la riabilitazione degli arti superiori richiede molte ore di allenamento quotidiano per diverse settimane, con il rischio di monotonia e calo di interesse e perdita di fiducia da parte dei pazienti [11].

La perdita di interesse dovuta alla natura ripetitiva degli esercizi può portare a una scarsa aderenza al programma riabilitativo, con conseguente riduzione dell'efficacia del trattamento e aumento del rischio di abbandono [11], [35]. La *Self-Determination Theory* (SDT) descrive la motivazione come fortemente influenzata dal contesto sociale e identifica tre bisogni psicologici di base: autonomia, competenza e interazione sociale [36]. La soddisfazione di questi tre bisogni facilita i processi di interiorizzazione, ovvero la trasformazione di una motivazione esterna ("lo faccio perché devo") in una motivazione più personale e autonoma ("lo faccio perché per me è importante") [36].

Pertanto, strutturare correttamente un esercizio in un sistema di teleriabilitazione risulta di primaria importanza. La pratica è più efficace quando è *task-specific*, cioè costruita su compiti significativi e rilevanti per le attività della vita quotidiana, e quando include un certo grado di variabilità, che permette di esplorare diverse soluzioni motorie e di adattarsi a contesti differenti. [36]. Allo stesso tempo, il livello di difficoltà deve essere calibrato sulle capacità del paziente: il compito deve essere abbastanza impegnativo da richiedere attenzione e *problem solving*, ma non così difficile da generare frustrazione. Il senso di progresso, il feedback sugli elementi chiave del movimento e la percezione di auto-efficacia favoriscono l'aderenza al trattamento e facilitano i processi di apprendimento motorio [32] [36].

L'impiego di *serious game* e tecnologie di realtà virtuale (VR) rappresenta una delle innovazioni più significative nello sviluppo della teleriabilitazione moderna. L'integrazione di elementi di gioco, quali feedback immediati, progressi visibili, sfide, punti e ricompense può migliorare in modo sostanziale la motivazione, l'aderenza e il coinvolgimento degli utenti nelle attività terapeutiche, soprattutto in contesti domiciliari asincroni [12].

Nel contesto dei *serious games* riabilitativi multiutente, Pereira et al. (2021) [37] distinguono diverse modalità di interazione:

- Competitiva: il successo del giocatore è confrontato con l'altro.
- Co-attiva o cooperativa: gli utenti giocano simultaneamente ma con ruoli differenti (dipendenza minima o nulla).
- Collaborativa: i giocatori condividono lo stesso obiettivo e dipendono l'uno dall'altro per completarlo.

In alcuni casi ai giochi competitivi sono associate *performance* più intense, maggior numero di ripetizioni e aumento del coinvolgimento, ma di contro possono causare un aumento dello stress, atteggiamenti aggressivi e risultare poco adatti a pazienti con capacità molto differenti [13], [37]. Dai risultati sperimentali di Pereira et al. (2021) [37] emerge che la modalità collaborativa promuove significativamente il *Behavioral Involvement*, cioè la percezione che le proprie azioni siano dipendenti da quelle degli altri giocatori. Gli autori interpretano questa dipendenza come positiva perché può favorire comunicazione e supporto reciproco. La modalità collaborativa emerge quindi come il paradigma più idoneo per promuovere l'interazione sociale e aumentare l'aderenza e l'efficacia del trattamento, riducendo al contempo il rischio di effetti indesiderati talvolta associati alle meccaniche competitive [13], [37].

La riabilitazione mediata dalla realtà virtuale (VR) permette di creare ambienti immersivi e controllati nei quali il paziente può esercitare movimenti dell'arto superiore attraverso compiti significativi e contestualizzati, spesso difficili da riprodurre nella pratica clinica tradizionale. La possibilità di simulare attività della vita quotidiana, unita alla natura interattiva della VR, offre condizioni ideali per stimolare i processi di apprendimento motorio e la neuroplasticità, elementi essenziali per il recupero funzionale dopo un danno neurologico [11]. Alcuni studi hanno documentato benefici anche in termini di autopercezione delle capacità e soddisfazione del paziente, suggerendo che la VR possa incidere positivamente non solo sul recupero motorio ma anche sugli aspetti psicologici che influenzano la partecipazione alla terapia. Inoltre, La VR si presta particolarmente bene alla creazione di ambienti multiutente in cui la condivisione dello spazio virtuale, l'animazione degli avatar e la comunicazione in tempo reale favoriscono un'interazione sociale più naturale [13].

Teleriabilitazione basata sulla Realtà Virtuale

In questa sezione verranno analizzate le tecnologie impiegate nei sistemi di teleriabilitazione basati su realtà virtuale, con particolare attenzione sui feedback forniti agli utenti.

Realtà Virtuale non immersiva

I primi sistemi di teleriabilitazione VR si basavano su realtà virtuale non immersiva, in cui i movimenti del paziente venivano riprodotti in uno schermo.

Burdea et al. (2000) [38] descrivono uno dei primi sistemi di teleriabilitazione ortopedica basata su realtà virtuale e interfacce aptiche. Il terapista e il paziente comunicano tramite due PC connessi via internet, accessoriati con telecamera e microfono. Per svolgere la sessione riabilitativa il paziente indossa il guanto Rutgers Master, in grado di misurare la posizione delle dita e le forze esercitate, e un sistema di tracciamento elettromagnetico posizionato sul polso. Il sistema fornisce sia feedback visivi, come la posizione della mano nell'ambiente virtuale, sia aptici attraverso il guanto. Successivamente sono stati sviluppati sistemi di tracciamento del movimento basati su telecamere 2D (*video-capture)*. In questo contesto Sucar et al. (2014) [39] propongono un sistema strutturato in cinque moduli principali: sistema fisico (PC, webcam e un controller con sfera tracciabile e sensore di pressione), sistema di *tracking* 3D basato su "*particle filter*" (ricostruzione 2D e stima della profondità), ambiente di gioco, modulo di rilevazione della compensazione del tronco e modulo di adattamento della difficoltà basato su modelli decisionali probabilistici (POMDP). La piattaforma propone tre minigiochi pensati per sollecitare movimenti di abduzione-adduzione e flesso-estensione della spalla, associati ad allenamenti sulla presa. La difficoltà, rappresentata da distanza fra i bersagli, tempo disponibile, velocità del bersaglio o soglia di pressione, viene modulata automaticamente dal POMDP. Al paziente vengono forniti feedback visivi e uditivi relativi alla posizione della mano e alla rilevazione di movimenti compensatori del tronco (bloccando il gioco o tramite allarme). I dati della sessione come la velocità e la *smoothness* della traiettoria.

L'introduzione di dispositivi commerciali a basso costo, come il Microsoft Kinect, dotati di telecamere RGB-D in grado di acquisire simultaneamente informazioni di colore e profondità, ne ha favorito l'ampia diffusione nelle piattaforme di teleriabilitazione. Una revisione della letteratura relativa al periodo 2015–2020 ha individuato nel Microsoft Kinect il dispositivo più utilizzato per il tracciamento dei movimenti nella teleriabilitazione dell'arto superiore [40]. Per esempio, Kato et al. (2015) [41] presentano un sistema di teleriabilitazione basato su realtà virtuale che combina Kinect e feedback vibrotattile per l'allenamento dell'arto superiore in pazienti post-ictus. Gli esercizi riabilitativi comprendevano movimenti di *reaching* verso l'alto, laterale e in avanti eseguiti in posizione seduta. La posizione dei bersagli nello spazio e i movimenti degli arti superiori del partecipante vengono mostrati su un display 3D. Il vibroattuatore, fissato alla mano, si attiva quando il paziente tocca il bersaglio virtuale, fornendo un feedback somatosensoriale aggiuntivo. Il terapista assiste alla sessione tramite videochiamata (mediante Skype) e può inviare dei comandi in tempo reale al paziente. I dati relativi agli angoli articolari, la posizione dei segmenti corporei e il tempo di completamento del task sono salvati in un database (Dropbox).

Adams et al. (2018) [42] hanno valutato l'efficacia di un sistema di realtà virtuale specificamente progettato per allenare attività di vita quotidiana (ADL) nella riabilitazione dell'arto superiore post-ictus, utilizzando SaeboVR. La piattaforma è costituita da PC, sensore Kinect e monitor ad alta definizione e può essere impiegato in maniera autonoma o sotto la supervisione di un operatore sanitario. Il sistema propone un "Road to Recovery" composto da 12 attività integrate di tipo ADL (es. fare la spesa, preparare pasti, curare gli animali, giardinaggio, organizzare armadi), con difficoltà regolabile dal terapista. Le scene richiedono sia componenti motorie (raggiungere, afferrare, spostare oggetti a differenti altezze e distanze) che cognitive (attenzione, memoria, pianificazione, sequenziamento). Il medico è in grado di regolare l'ampiezza di movimento e il numero di ripetizioni necessarie per completare le attività adeguando il livello di difficoltà alle capacità del paziente. In Figura 1 sono riportati alcuni esercizi proposti. I movimenti del paziente nel mondo reale sono replicati nel mondo virtuale tramite un avatar, in aggiunta, sono presenti feedback uditivi da parte del "virtual therapist" e un sistema di ricompensa quando il soggetto completa il task.

Figura 1: Esempi di ADL presenti nella piattaforma SaeboVR. In alto a sinistra: preparare i pasti, in alto a destra: giardinaggio, in basso a sinistra consumare un pasto, in basso a destra: fare il bucato. Tratta da [42]

Trombetta et al. (2017) [43] hanno realizzato una piattaforma denominata Motion Rehab AVE 3D progettata per la riabilitazione motoria degli arti superiori, inferiori e dell'equilibrio in pazienti affetti da esiti di ictus lieve. L'architettura supporta diverse modalità di visualizzazione (TV o HMD come l'Oculus Rift). Il protocollo riabilitativo integrato nel software prevede l'esecuzione di sei diverse attività in un ambiente tridimensionale, durante le quali il paziente controlla un avatar per interagire con oggetti virtuali. I task richiedono l'esecuzione di specifici gesti terapeutici che coinvolgono movimenti dell'arto superiore e inferiore. Durante le sessioni, il paziente riceve feedback visivi istantanei riguardanti il punteggio, la correttezza dell'azione eseguita e messaggi motivazionali, accompagnati da stimoli sonori di allerta o gratificazione associati al completamento dei task. Il sistema registra inoltre i successi e gli errori commessi durante gli esercizi, fornendo al fisioterapista dati quantitativi utili per valutare l'evoluzione del

recupero motorio del paziente. Yang et al.(2022) [44] propongono AR Rehab (Figura 2), una piattaforma di riabilitazione in realtà aumentata. Il sistema, installato su carrello mobile, utilizza la camera RGB-D del Kinect V2 per tracciare uno scheletro 3D del paziente e fornisce feedback visivi in tempo reale su posture, traiettorie, distanze di *reaching* e angoli articolari. Le esercitazioni (46 in totale) comprendono esercizi per arto superiore, arto inferiore, equilibrio e ADL. Il terapista seleziona gli esercizi dalla libreria, imposta la difficoltà e durata della sessione sulla base della valutazione iniziale, monitora il *training* da remoto e riceve report con cadenza settimanale, usando tali informazioni per modificare la prescrizione in base all'andamento del paziente. Il sistema genera un report immediato al termine di ciascun esercizio con dettagli prestazionali e confronto con le esecuzioni precedenti; inoltre registra e sincronizza sul *cloud* i video dello scheletro 3D rivedibili dal paziente e dal terapista per individuare le fasi più critiche dell'esecuzione.

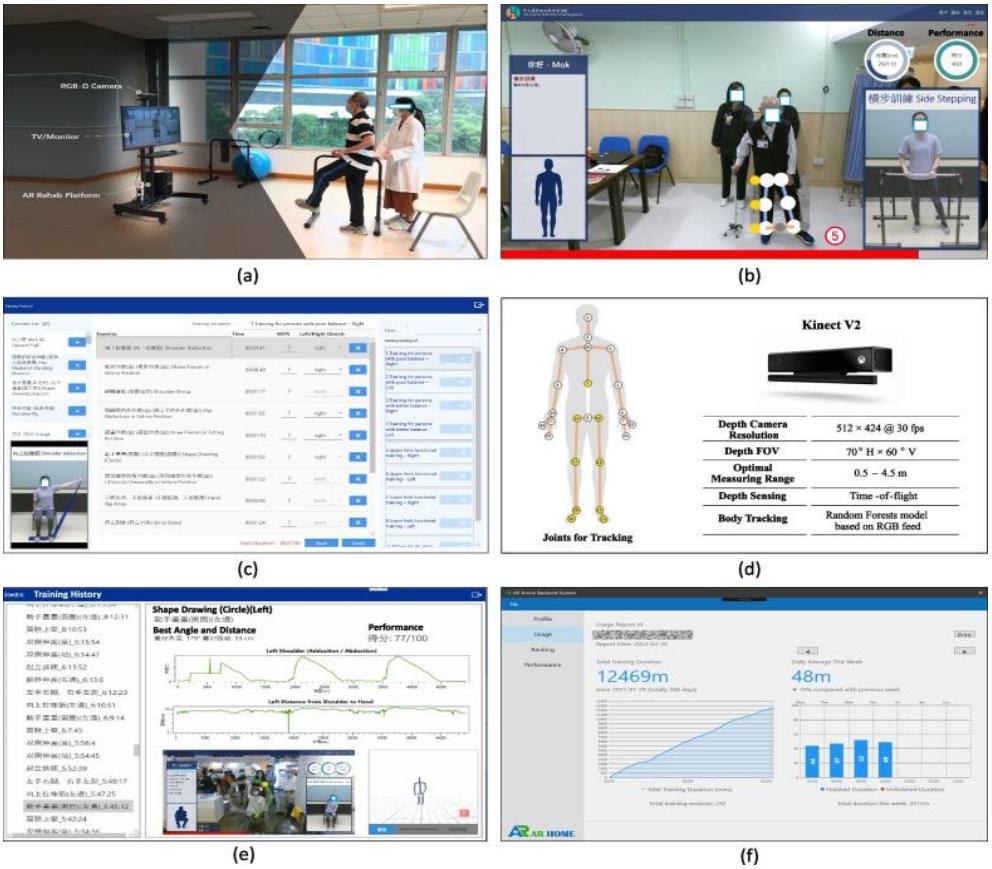

Figura 2: Il sistema AR Rehab per l'allenamento virtuale. a: Architettura del sistema e scena di allenamento; b: Allenamento in corso; c: Personalizzazione del piano di allenamento; d: Articolazioni per il tracciamento del corpo e sensore Kinect; e: Report in tempo reale; f: Statistiche di allenamento. Tratta da [44]

Le telecamere RGB-D presentano varie criticità nella stima della posizione: soffrono di problemi di occlusione, dipendenza dalle condizioni di luce, errori nell'algoritmo di ricostruzione e risoluzione limitata nel piano trasversale se non posizionate correttamente. Una tecnologia alternativa per il tracciamento dei movimenti dell'arto superiore sono le unità inerziali (IMU) che, per determinare l'orientamento del sensore nello spazio, integrano accelerometri e giroscopi, spesso corredati anche da magnetometri (MIMU). I MIMU rappresentano una soluzione valida, nonostante richiedano un set-up più complesso e procedure di calibrazione accurate [45].

Questa tecnologia è impiegata, ad esempio, nel sistema DoMoMEA [46], una piattaforma di teleriabilitazione *home-based* per pazienti post-ictus con deficit lieve. Il sistema utilizza 7 MIMU per la ricostruzione della cinematica articolare (tronco e arti) e sensori FSR (*force-sensing resistors*) per acquisire la pressione di contatto nella soletta della scarpa, nei gomiti e tra il pollice e l'indice. I sensori sono fissati sul corpo mediante fasce in neoprene con un codice a colori specifico (Figura 3), inoltre la piattaforma integra procedure guidate di configurazione e calibrazione per rendere la vestizione fattibile anche a domicilio. Gli *exergame* proposti includono movimenti per gli arti superiori (abduzione/adduzione di spalla, flesso-estensione di gomito e polso), inferiori (flesso-estensione di anca, ginocchio e caviglia) e del tronco (rotazione e flesso-estensione del tronco). I parametri biomeccanici estratti (ROM, tempo di esecuzione, numero di ripetizioni, presenza di movimenti compensatori) sono usati sia per fornire biofeedback visivi e uditivi al paziente, sia per il telemonitoraggio asincrono da parte del terapista, che tramite una web app può programmare le sessioni e adattare la difficoltà degli esercizi in base ai progressi, garantendo così un allenamento sicuro e stimolante.

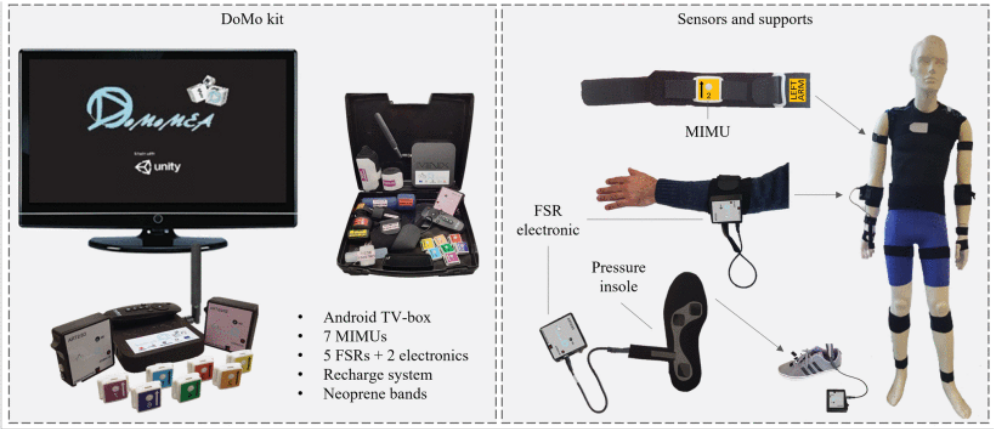

Figura 3: Il kit DoMo del sistema DoMoMEA, include un Android TV-box, sette MIMU, cinque sensori FSR con due dispositivi elettronici di lettura personalizzati, un sistema di ricarica e una serie di fasce in neoprene per fissare i sensori sul corpo. Tratta da [46] .

Una piattaforma a basso costo è stata implementata da Tsekleves et al. (2016) [47]. Il sistema utilizza due controller Wii posizionati sul braccio e avambraccio per stimare con una buona accuratezza la posizione dei segmenti corporei e animare l'arto di un avatar all'interno del gioco. La piattaforma include giochi riabilitativi adattabili per velocità, ROM e difficoltà, con una modalità di "*virtual teacher*" per istruire il paziente e un'applicazione per il terapista funzionale alla programmazione e monitoraggio delle sessioni da remoto.

Per la riabilitazione dei segmenti più distali dell'arto superiore, possono essere impiegati specifici guanti sensorizzati (Figura 4 A) per tracciare i movimenti della mano e delle dita per permettere l'interazione del soggetto con l'ambiente virtuale effettuando la presa e il rilascio di oggetti [48], [49].

Un'alternativa commerciale per la rilevazione dei movimenti della mano è la Leap Motion Controller (LMC) (Figura 4 B). Questa tecnologia è stata utilizzata nella piattaforma HoVRS (Home based Virtual Rehabilitation System) proposta da Qiu et al. (2020) [50]. Gli autori hanno sviluppato vari giochi calibrati sul ROM residuo del paziente, che adattavano dinamicamente la difficoltà in base alla performance, fornendo biofeedback visivi in tempo reale (movimento dell'avatar, traiettorie da seguire, punteggi) e report automatici per il monitoraggio del terapista da remoto.

Figura 4: Esempio di sistemi per il tracciamento dei movimenti della mano. A: Ortesi SaeboGlove con sensori per monitorare le interazioni di presa, tratta da [48]. B: Leap Motion Controller (LMC) con supporto passivo per il braccio, tratta da [50].

Per i pazienti con compromissione motoria severa, dove il movimento funzionale è assente o minimo, le piattaforme basate esclusivamente sul tracciamento del movimento o su compiti funzionali in realtà virtuale risultano poco adatte. In questi casi i sistemi di teleriabilitazione sfruttano l'attività muscolare residua come sorgente primaria di biofeedback. Un esempio è rappresentato da Tele-REINVENT [51], una piattaforma *home-based* composta da un laptop, sensori EMG *low-cost* posizionati sui muscoli flessori ed estensori del polso e un'interfaccia software dedicata e una serie di giochi adattivi (es. Skeeball, Planet Jump, Blinko). Prima di ogni sessione viene eseguita una calibrazione EMG giornaliera per adattare le soglie dei giochi al livello di attivazione muscolare di quel giorno, in modo che anche piccoli tentativi di estensione/flessione del polso possano animare l'avatar o controllare gli eventi di gioco. In questo paradigma, il segnale EMG viene trasformato in biofeedback visivi e nel controllo diretto del gioco, rinforzando il tentativo di movimento anche in assenza di spostamento articolare misurabile, i modo da promuovere la consapevolezza e il controllo dell'attività muscolare residua. Le sessioni vengono svolte a domicilio, con sedute supervisionate in teleconferenza (per supporto tecnico, aggiustamento dei parametri e correzione posturale) e autonome. Nei sistemi multimodali l'attività EMG può essere integrata con sistemi di rilevamento dello sguardo (*eye tracker*) per rilevare in manera più accurata l'intenzione del movimento e dispositivi aptici per fornire un biofeedback tattile sul completamento del task in aggiunta a quelli visivi e uditivi [52].

Realtà Virtuale immersiva

A differenza dei sistemi non immersivi, che proiettano l'ambiente virtuale su uno schermo esterno e mantengono il paziente consapevole del contesto reale circostante, i sistemi di realtà virtuale immersiva hanno l'obiettivo di sostituire completamente il mondo reale con uno spazio digitale tridimensionale. Attraverso l'uso di visori (*Head Mounted Display* - HMD) e sensori di tracciamento, il paziente viene collocato all'interno dell'ambiente terapeutico, sperimentando una sensazione di presenza molto più forte. Questa caratteristica permette al paziente di interagire in prima persona con oggetti e scenari riabilitativi virtuali eseguendo eseguire movimenti più naturali e completi. La realtà virtuale immersiva offre una maggior stimolazione sensoriale, aumentando il senso di presenza e l'impegno, rendendo l'esercizio riabilitativo più coinvolgente [53]. In letteratura sono stati proposti diversi sistemi di teleriabilitazione degli arti superiori basati su realtà virtuale immersiva.

Un esempio è CogniviveVR [54], un sistema *home-based* per il recupero dell'arto superiore che utilizza giochi terapeutici adattivi come *Rally Zone*, dove il paziente indossa il visore Oculus Rift e tramite i controller muove una racchetta virtuale per colpire una palla contro blocchi colorati. I movimenti richiesti includono *forward reach* (colpire palla servita davanti alla spalla), *lateral reach* (deviazioni laterali e controlaterali), *vertical reach* (movimenti in elevazione). Il sistema applica un *Dynamic Difficulty Adjustment* (DDA) che adatta in tempo reale distanza, altezza e velocità della palla al ROM attivo del paziente, stimolando i limiti funzionali del soggetto. Le metriche di performance monitorate includono SPARC (*Spectral Arc Length*), un indice di fluidità del movimento che quantifica l'efficienza cinematica. Il terapista visualizza tramite dashboard web l'aderenza terapeutica (frequenza/durata sessioni), il numero ripetizioni e la SPARC, consentendo il monitoraggio e l'aggiornamento della terapia da remoto.

Lo studio di Chatterjee et al. (2022) [55] introduce VIRTUE, una piattaforma dedicata specificamente alla riabilitazione cognitiva post-ictus (*Post-Stroke Cognitive Impairment* - PSCI). Utilizzando un visore Oculus Rift S e controller manuali, il sistema immerge il paziente in ambienti realistici (cucina, camera da letto, bagno, ristorante, giardino). I compiti proposti simulano fedelmente le ADL, ma con un focus sulla sequenzialità e sul *problem solving*, ad esempio preparare un pasto, scegliere i vestiti, ordinare in un ristorante, pagare il conto. Le attività sono svolte in posizione seduta e con il braccio non paretico, mentre la difficoltà è scalabile. Il terapista può vedere in tempo reale la visuale del paziente, personalizzare la sessione, modificare il livello di difficoltà o comunicare con esso mentre è immerso nella scena. In VIRTUE, il feedback non è volto alla correzione del gesto motorio, ma al rinforzo cognitivo. Il sistema fornisce istruzioni audio e testuali per guidare il paziente e offre un feedback immediato sul completamento corretto della sequenza logica del compito.

Sebbene i sistemi presentati abbiano riscontrato effetti positivi sia in termini di risultati clinici che di accettazione, l'interazione con l'ambiente virtuale eseguita direttamente con le mani dell'utente, senza l'utilizzo dei controller, potrebbe fornire un'esperienza più naturale, coinvolgente e motivante [56].

Per integrare questa modalità, una soluzione è rappresentata dall'utilizzo della Leap Motion, attaccandola frontalmente al visore. Questa configurazione è stata validata in piattaforme di riabilitazione immersiva VR per compiti di *reaching* [57], *reach-to-grasp* mono o bimanuali [58], manipolazione di oggetti e ADL [59], seppur con applicazioni ancora limitate al contesto clinico e poche evidenze di implementazione in teleriabilitazione remota.

Accanto alle soluzioni *controller-free* basate su HMD e sensori esterni, sono stati sviluppati visori in cui il tracciamento della mano è incorporato nel dispositivo (*inside-out hand-tracking*).

Un esempio è il sistema proposto da Fregna et al. (2022) [60], basato su Meta Quest 2 e un'applicazione per PC che permette al terapista di monitorare in tempo reale il paziente e, se necessario, interagire con lui vocalmente (Figura 5 A e B). I task proposti dall'applicazione sono quattro: *Ball in hole, Cloud, Glasses, Rolling Pin*, progettati per allenare *reaching*, puntamento, coordinazione e manipolazione (Figura 5 C). Durante l'esecuzione dei task, il paziente è seduto di fronte ad un tavolo e i movimenti delle mani e delle dita, catturati visivamente dal software integrato del visore, sono utilizzati per animare due mani virtuali. Inoltre, il tavolo virtuale viene registrato su quello reale, introducendo un riscontro tattile/propriocettivo congruente con il mondo reale. Il Meta Quest 2 è stato utilizzato anche da Amin et al. (2024) [61] per la riabilitazione della mano. In questo caso i task sono progettati per allenare movimenti specifici come flessione/estensione del polso (Hit a Rolling Ball*)*), apertura/chiusura con mantenimento della presa (*Grasp a Balloon)*, supinazione/pronazione (*Swap Hand*), presa a pinza (*Grip a Pencil*). Anche in questo caso il feedback visivo è rappresentato dalla mano virtuale, ed è arricchito da una componente di *cognitive engagement*: oggetti con colori da selezionare, richiesta di attenzione e *decision-making* (scegliere l'oggetto del colore corretto), esito del compito e una *scoreboard* (tempo, successi/insuccessi).

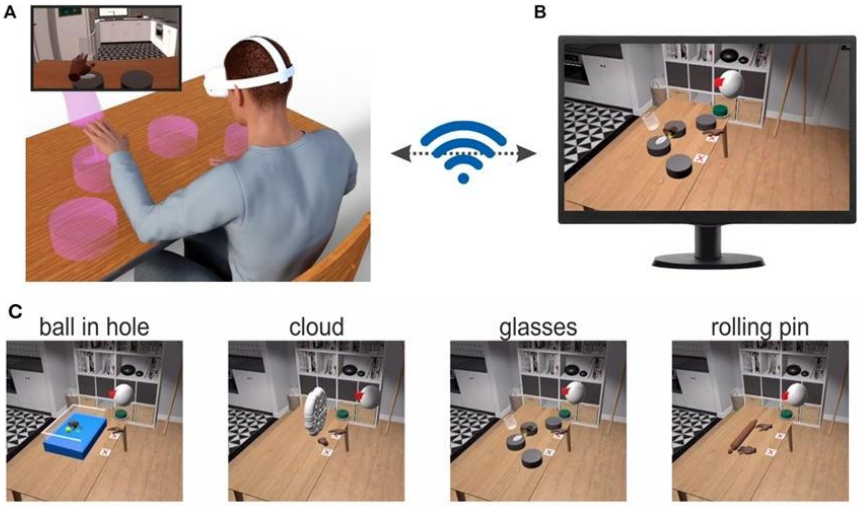

Figura 5: Scenario applicativo dell'ambiente VR immersivo sviluppato da [60] (A) L'immagine mostra la scena così come viene percepita dal paziente sul display dell'HMD. (B) App client in esecuzione su un PC che consente di monitorare in remoto e in tempo reale il comportamento dei pazienti, impostare le loro routine di riabilitazione e interagire vocalmente con loro. (C) Le quattro attività proposte dal sistema. Tratta da [60]

Una piattaforma recente, denominata RecoveryFun, è stata proposta da Benadduci et al (2025) [62]. Il sistema utilizza il visore PICO 4 (tracciamento delle mani e dello sguardo integrato) e un sensore indossabile che misura la risposta galvanica della pelle e la fotopletismografia per stimare stress e fatica durante la sessione riabilitativa. La piattaforma, mostrata in Figura 6, comprende un'applicazione per il caregiver, nella quale quest'ultimo può mandare messaggi motivazionali, monitorare l'aderenza alla terapia e consultare il materiale di supporto sull'uso del sistema (video esplicativi e sezione FAQ) per aiutare il paziente nell'utilizzo dei dispositivi, fungendo da assistenza di primo livello. L'assistente ha anche la possibilità di utilizzare il tablet durante la sessione riabilitativa del paziente per visualizzare l'ambiente VR in cui sta giocando. I professionisti (assistente, terapista e personale di gestione) si affidano alla piattaforma clinica, un'applicazione web, per gestire le impostazioni della sessione e monitorare i dati del paziente. Nella piattaforma un medico autorizzato può modificare i dati clinici e amministrativi dei pazienti assegnati, associare l'assistente al paziente corrispondente, creare piani di riabilitazione specifici, monitorare l'attività dei pazienti (progressi, numero e durata delle sessioni eseguite, di allenamento eseguite) ed eventualmente effettuare videochiamate. Gli esercizi proposti comprendono *exergame* motori e cognitivi (es schiaccia la talpa, labirinto, spesa, basket, riproduzione di gesti della mano, memory), parametrizzati tramite una fase di calibrazione iniziale.

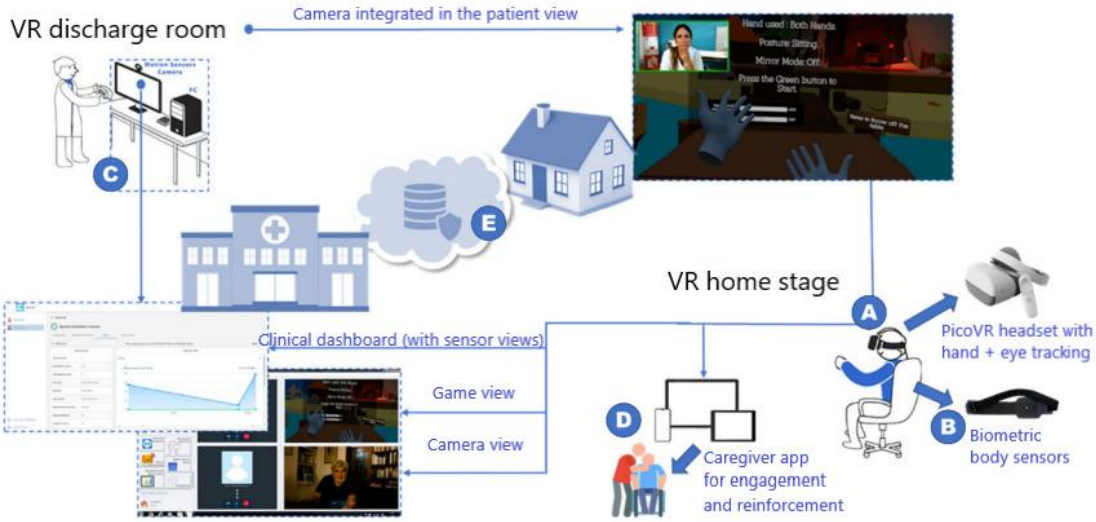

Figura 6:Architettura del sistema RecoveryFun: (A) visore per realtà virtuale, (B) sensore indossabile, (C) piattaforma clinica, (D) app per caregiver, (E) smart services. Tratta da [62].

Per quanto concerne i sistemi multimodali, l'integrazione della componente aptica può contribuire a migliorare l'*embodiment* e la coerenza sensoriale nell'ambiente immersivo. In letteratura, il feedback aptico viene realizzato mediante soluzioni tecnologiche eterogenee. Ad esempio, Zhang et al. (2024) [63] propongono un approccio passivo tramite il dispositivo TouchMark, che introduce rilievi tattili (ad es. linee o anelli) sull'impugnatura del controller. Tali caratteristiche sono poi associate agli oggetti virtuali impiegati nei task riabilitativi e in attività di tipo ADL (ad esempio gli utensili da cucina), fornendo un feedback sensoriale senza attuazione.

Una soluzione attiva è rappresentata dal feedback vibrotattile. Nel sistema basato su *rhythm game* per la riabilitazione della mano, gli autori impiegano vibroattuatori, posizionati sui polsi, come rinforzo multisensoriale dei feedback visivi e uditivi [64]. Durante il task l'utente deve eseguire ripetutamente dei gesti target sincronizzandosi con ritmo imposto dal sistema. Il feedback aptico segnala l'esecuzione corretta dell'azione.

Applicazioni più complesse utilizzano guanti aptici in grado sia di fornire una vibrazione complessa sia di registrare gli angoli articolari della mano [65]. Un'ulteriore estensione del paradigma vibrotattile è costituita dai sistemi multimodali che combinano vibrazione e stimolazione termica tramite dispositivi indossabili applicati alle dita [66]. Infine, oltre alla vibrazione, sono stati proposti dispositivi aptici indossabili in grado di fornire un feedback più informativo attraverso la deformazione controllata della pelle: tali interfacce applicano una pressione al polpastrello in corrispondenza dell'interazione con l'oggetto virtuale [67].

L'integrazione dell'attività muscolare è stata impiegata nel sistema di *Motor Imagery* (MI) sviluppato da Lin et al. (2023) [68], il biofeedback EMG è utilizzato per tradurre lo sforzo del paziente in un elemento di gioco tramite l'assegnazione di un punteggio maggiore se viene registrata un'attività muscolare più intensa.

La piattaforma *Handly* [69] propone un approccio ibrido, dove l'EMG lavora in sinergia con il tracciamento ottico delle mani (tramite Leap Motion). L'azione virtuale (es. afferrare un oggetto) viene eseguita solo se il segnale EMG supera la soglia personalizzata per almeno 250 ms, assicurando che il movimento tracciato otticamente sia accompagnato da una reale attivazione muscolare volontaria.

Nello studio di Li et al. (2019) [70], l'EMG funge da interfaccia muscolo-computer per sostituire completamente i controller manuali. Attraverso degli algoritmi di classificazione (come SVM e PNN) vengono identificati quattro tipologie di movimento: flessione/estensione del polso, chiusura e apertura della mano. Ad ogni movimento sono associate delle ADL nel mondo virtuale ed esempio aprire la porta, pulire il tavolo e tagliare il cibo.

Lo studio di Shih et al**.** (2024) [71] propone un'architettura che oltre ai biofeedback visti in precedenza (posizione nel mondo virtuale, feedback di performance e reportistica, *cognitive engagement*) integra il monitoraggio EEG per valutare lo stato emotivo del paziente. Inoltre, la piattaforma utilizza un sistema per rilevare posture anomale, strategie compensatorie e sincronizzazione degli arti nella *Mirror Therapy* (MT) (Figura 7 C), avvisando il paziente tramite messaggi e codice a colore.

Il protocollo riabilitativo include diversi esercizi *single-player* per la terapia occupazionale (Figura 7 A) , *Constraint-Induced Movement Therapy* (CIMT) in cui viene limitato l'utilizzo dell'arto non paretico (Figura 7 B) e MT (Figura 7 C). Un elemento distintivo di questa piattaforma è l'integrazione della modalità *multiplayer*, in cui il terapista ha la possibilità di accedere al mondo virtuale e far svolgere al paziente giochi cognitivi per allenare velocità di reazione, memoria ed equilibrio (Figura 7 D). Il segnale EEG acquisito tramite apposito caschetto viene visualizzato in tempo reale nell'ambiente virtuale tramite mappa di attivazione delle aree corticali. Il terapista è in grado di valutare se il paziente è ansioso, frustrato o distratto ed eventualmente regolare manualmente l'intensità o la difficoltà del compito per mantenere uno stato emotivo ottimale per l'apprendimento.

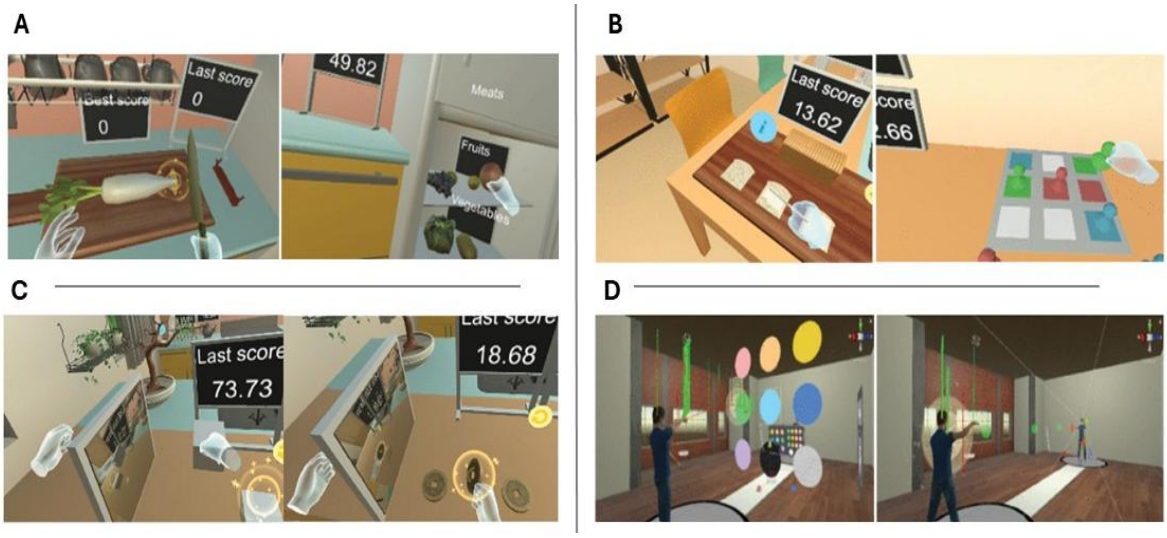

Figura 7: Esercizi proposti dalla piattaforma sviluppata da Shih et al. (2024) [71]. (A) Terapia Occupazionale: a sinistra estensione del gomito/avambraccio, a destra abduzione e adduzione orizzontale della spalla. (B) CIMT: a sinistra flessione ed estensione del gomito, a destra rotazione esterna e interna della spalla. (C) MT: a sinistra supinazione e pronazione del polso, a destra supinazione e pronazione del gomito. (D) Allenamento cognitivo in multiplayer: a sinistra allenamento della velocità di reazione, a destra allenamento della memoria e del senso dell'equilibrio. Immagini tratte da [71].

Dall'analisi della letteratura emerge che gli approcci multiutente in realtà virtuale con meccaniche collaborative risultano poco esplorati in ambito riabilitativo. Tuttavia, alcuni studi hanno investigato tale combinazione in popolazioni anziane e in pazienti post-ictus, evidenziando un incremento della prestazione [72], della motivazione intrinseca e del coinvolgimento [13]. Il presente lavoro di tesi si inserisce in questo contesto, proponendo un sistema di biofeedback a supporto di un *exergame* collaborativo in realtà virtuale per la teleriabilitazione dell'arto superiore.

Movimenti compensatori durante il reaching

In questa sezione verrà presentato il movimento di *reaching* per la riabilitazione degli arti superiori nei pazienti post ictus. Verranno inoltre descritte le strategie compensatorie adottate dai pazienti durante l'esecuzione del movimento, i metodi per rilevarle e i relativi feedback.

Introduzione

A seguito di una lesione cerebrale i deficit motori sono molto comuni e possono manifestarsi come debolezza muscolare, perdita di coordinazione, spasticità, alterazione delle sinergie muscolari e della sensibilità, rallentamento motorio, ridotta selettività e destrezza del movimento, condizioni che portano il paziente ad abbandonare progressivamente l'utilizzo dell'arto menomato [73]. Il mancato utilizzo porta al *learned non-use*, ovvero il soggetto non utilizza l'arto durante le attività di vita quotidiana nella fase successiva al trauma nonostante potenzialmente potrebbe svolgere alcune attività. L'immobilità a sua volta può innescare un circolo vizioso di problemi, tra cui alterazioni dei tessuti molli periferici, cambiamenti nella densità minerale ossea, potenziamento dei meccanismi riflessi e spasticità, portando infine alla fibrosi muscolare e contribuendo a posture anomale degli arti, dolore e diminuzione della funzionalità [73]. Per contrastare questo declino e interrompere il circolo vizioso del disuso, è necessario implementare programmi riabilitativi specialmente nella fase immediatamente successiva all'ictus (3-6 mesi), detta "finestra critica", in cui i meccanismi di neuroplasticità spontanea e riparazione biologica sono massimi [74]. Tuttavia, il potenziale riabilitativo non si esaurisce con la fase subacuta. Ward et al. (2019) [75] hanno dimostrato che anche a distanza di anni dall'evento è possibile ottenere miglioramenti clinici significativi. Attraverso un programma di neuroriabilitazione ad alto dosaggio e alta intensità, è stato osservato che i pazienti cronici possono ridurre la menomazione e migliorare l'attività funzionale.

Biomeccanica del reaching e analisi delle compensazioni dei pazienti post-ictus

Il movimento di *reaching* consiste nel posizionamento volontario della mano in corrispondenza o in prossimità di un punto nello spazio per poter interagire con l'ambiente. Questo movimento rappresenta una componente centrale di quasi tutte le attività della vita quotidiana, ad esempio: cura personale, alimentazione, igiene, spostamenti, vestizione. Si tratta di un compito funzionale multiarticolare, che coinvolge spalla, gomito, polso e per bersagli oltre la lunghezza del braccio, anche tronco e anca. Il controllo neuromuscolare del movimento di *reaching* è complesso e richiede la sincronizzazione dell'attivazione muscolare in tutte le articolazioni in movimento, nonché di tutti i muscoli coinvolti nella stabilizzazione posturale. L'accelerazione di ciascuna articolazione durante un movimento di *reaching* dipende sia dalla coppia articolare netta (cioè dalla forza di rotazione) sia dall'inerzia (resistenza di un oggetto a qualsiasi cambiamento di movimento). Per controllare questo movimento il sistema nervoso centrale (SNC) utilizza strategie sia di *feedforward* (pianificazione anticipatoria) e feedback (correzione in corso d'opera). Nonostante la complessità del movimento, le caratteristiche cinematiche sono altamente riproducibili nei pazienti sani, manifestate in traiettorie della mano tendenzialmente rettilinee e profili di velocità "a campana". Nel *reaching* i pazienti post ictus presentano traiettorie che si discostano notevolmente rispetto ai soggetti sani, risultano più lente, meno fluide e precise, con movimenti segmentati per raggiungere l'oggetto [76] (Figura 8).

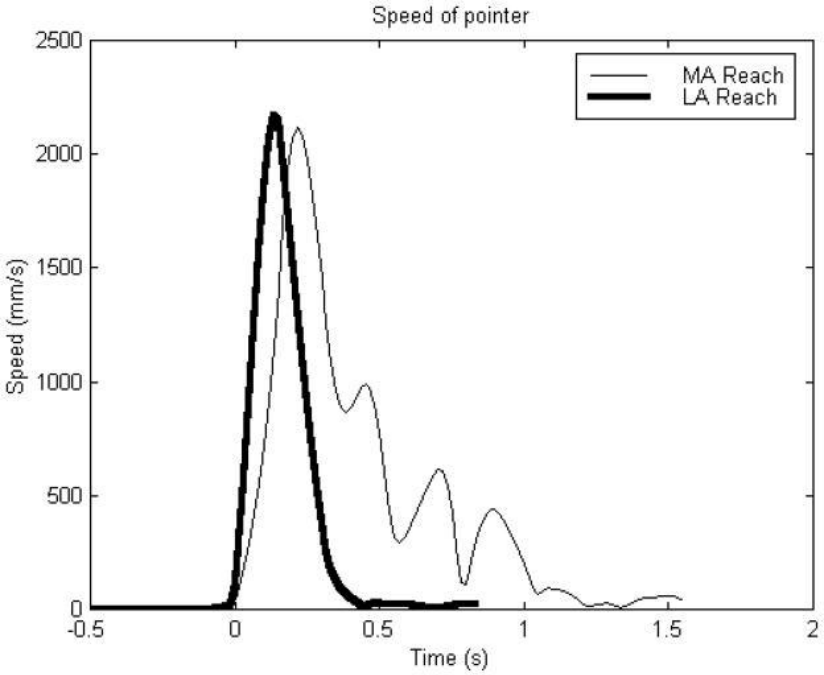

Figura 8: Profilo di velocità della mano in un paziente post ictus per il braccio più colpito (MA) e quello meno colpito (LA) durante un compito di reaching. Il soggetto (età=57, lesione=ictus emorragico lato destro, tempo trascorso dall'ictus=6 anni, funzione degli arti superiori secondo Fugl Meyer=19/66) era seduto con una cintura sul torace per limitare il movimento del tronco. Un segnale acustico indicava al soggetto di raggiungere il più rapidamente possibile un bersaglio all'altezza delle spalle partendo da una posizione iniziale a metà coscia. Tratta da [76].

A livello biomeccanico i deficit motori si riflettono in coordinazione inter-articolare alterata, ridotta estensione del gomito, ridotta protrazione scapolare, minore rotazione laterale dell' articolazione gleno-omerale, una minor flessione e abduzione orizzontale della spalla con conseguente adozione di strategie compensatorie [77], [78].

La compensazione motoria viene definita da Levin et al. (2009) [79] come l'emergere di nuovi *pattern* (schemi) motori risultanti dall'adattamento degli elementi residui o dalla sostituzione funzionale, intesa come l'assunzione di funzioni motorie da parte di segmenti corporei alternativi. Possono distinguersi due tipologie di compensazione:

- adattiva**:** si riferisce all'uso di *pattern* di movimento alternativi per eseguire un compito.
- sostitutiva**:** implica l'uso di effettori diversi per sostituire gli elementi motori perduti.

Un esempio classico di compensazione sostitutiva è rappresentato dalla rotazione e flessione, sia sul piano sagittale che laterale, del tronco [80], [81]. Cirstea e Levin (2000) [80] hanno dimostrato che i pazienti post-ictus reclutano questi gradi di libertà aggiuntivi non solo per estendere il raggio d'azione, ma anche per compiti all'interno dello spazio raggiungibile, correlando questo comportamento alla gravità del deficit motorio e alla ridotta escursione attiva di gomito e spalla.

Per quanto concerne la compensazione adattiva, un esempio significativo è l'elevazione della spalla. Questo movimento viene utilizzato per sollevare la mano al di sopra dell'oggetto target, compensando così una ridotta flessione della spalla e una debolezza dei muscoli distali che impedisce un'adeguata stabilizzazione del polso e delle dita [82], [83]. Anche l'abduzione della spalla è frequentemente adottata come strategia compensatoria durante varie tipologie di *reaching* [81], [82], [84]. Nel *forward reaching* questo compenso, associato all'intra-rotazione del gomito, comporta un approccio laterale al target piuttosto che frontale [85] (Figura 9).

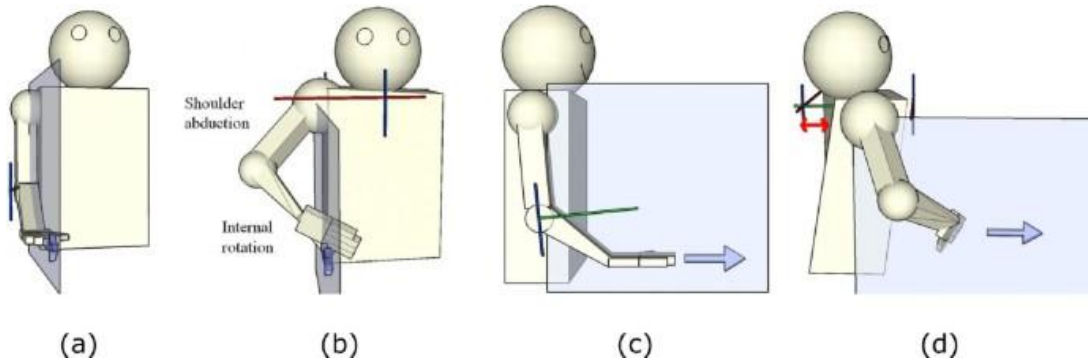

Figura 9: Strategie compensatorie durante l'esercizio di reaching. *I pazienti che hanno subito un ictus spesso presentano abduzione della spalla, intra-rotazione del gomito e rotazione del tronco anomale durante l'esecuzione del compito. (a) Vista frontale dell'estensione normale, (b) vista frontale dell'estensione anomala, (c) vista laterale dell'estensione normale e (d) vista laterale dell'estensione anomala. Tratta da [85].*

Queste alterazioni cinematiche sono spesso guidate dalle sinergie flessorie patologiche. Levin et al. (2016) [81] descrivono questa caratteristica come un accoppiamento stereotipato tra i movimenti della spalla e del gomito, quantificabile attraverso il movimento del "piano del braccio" (piano descritto dai segmenti che uniscono spalla gomito e mano). È emerso che nei pazienti con compromissione lieve, l'uso della sinergia è adattabile in base al compito richiesto e contribuisce all'accuratezza del *reaching*, permettendo ad esempio di sollevare il braccio per compensare l'abbassamento della mano. Al contrario, nei pazienti con compromissione moderata o grave, questo pattern diventa fisso e stereotipato, perdendo la sua natura adattiva e costringendo il paziente ad ulteriori compensi del tronco per orientare la mano.

L'impatto di queste sinergie sull'area esplorabile dall'arto patologico, definita *work area,* è stato indagato da Sukal et al. (2007) [86]. Utilizzando un dispositivo robotico per imporre carichi progressivi durante l'abduzione della spalla, gli autori hanno dimostrato che l'area di lavoro si riduce drasticamente all'aumentare del peso che deve sostenere il braccio paretico. Questo fenomeno è attribuito all'incapacità di vincere l'accoppiamento obbligato tra abduzione della spalla e flessione del gomito che ne inibisce l'estensione.

Ota et al. (2023) [84] mediante analisi cinematica 3D, hanno quantificato i movimenti compensatori del tronco durante il *forward reaching* e della spalla nell' *hand-to-mouth reaching* (portare la mano alla bocca). I risultati mostrano un contributo della flessione della spalla e del gomito significativamente ridotto nei pazienti rispetto ai sani, mentre aumentano la flessione del tronco, l'elevazione del cingolo scapolare e l'abduzione della spalla (Figura 10 A e B).

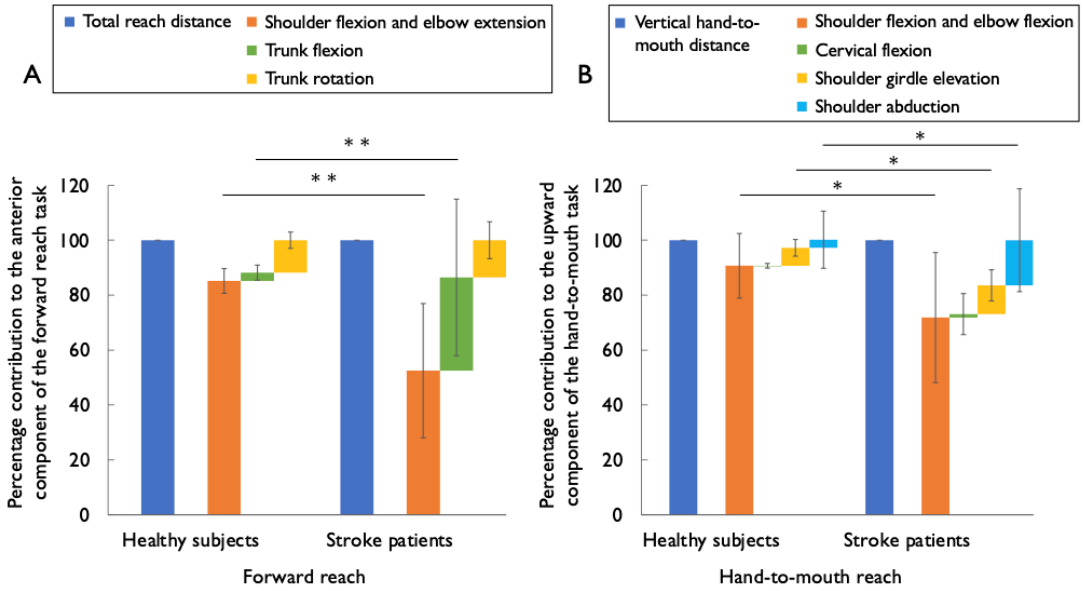

*Figura 10: Contributi cinematici durante il forward e hand-to-mouth reaching nei pazienti sani e patologici. (A) Distanza totale raggiunta nei movimenti in avanti sull'asse Y (blu) e le sue componenti in partecipanti sani e partecipanti con ictus. Componenti: flessione della spalla o estensione del gomito (arancione), flessione del tronco (verde) e rotazione del tronco (giallo). (B) Distanza verticale mano-bocca sull'asse Z (blu) e sue componenti in partecipanti sani e partecipanti con ictus. Componenti: flessione della spalla o estensione del gomito (arancione), flessione cervicale (verde), elevazione del cingolo scapolare (giallo) e abduzione della spalla (azzurro). *P<.05, **P<.01. Tratta da [84].*

Infine, a livello distale, la mano paretica sviluppa strategie compensatorie specifiche a causa della perdita di forza e controllo delle dita, tendenza a flettere dita e polso, compromissioni e ritardi nella modulazione della forza di presa [87]. Roby-Brami et al. (2021) [87] descrivono strategie alternative come lo "*sliding*", far scivolare la mano sul tavolo verso l'oggetto per sostenerne il peso, o il "*grasping from above*" approccio verticale all'oggetto, e configurazioni di presa più grossolane per stabilizzare l'oggetto, a scapito di quelle di precisione. Xie et al. (2022) [82], analizzando il compito di *reach-to-grasp-pen*, hanno rilevato che i pazienti mostrano un'apertura della mano maggiore rispetto ai controlli e una ridotta capacità di modulare l'estensione dell'indice, suggerendo un deficit nel controllo fine necessario per scalare l'apertura della mano sulle dimensioni dell'oggetto.

Compresi i meccanismi di compensazione è necessario analizzare quali sono le loro implicazioni a livello clinico e riabilitativo. Sebbene l'adozione di strategie compensatorie possa garantire un successo iniziale nel completamento del compito, specialmente nelle fasi precoci o nei pazienti con deficit severi, questa modalità si rivela spesso controproducente nel lungo termine. La letteratura definisce questo fenomeno come *learned bad-use* (cattivo uso appreso): il

rinforzo positivo dato dal successo occasionale nel raggiungere l'obiettivo consolida *pattern* motori anomali a scapito del recupero di quelli fisiologici [73]. Pertanto seppure ci sia un miglioramento nell'attività, ovvero il completamento del un compito, non avviene un vero recupero funzionale, definito come il ripristino della capacità di eseguire un movimento nello stesso modo in cui veniva eseguito prima della lesione [79]. D'altro canto il modello motorio atipico utilizzato dal paziente può rappresentare un adattamento valido per migliorare le capacità fisiche residue, specialmente nei soggetti con ictus moderato [81]. È quindi importante fare una valutazione paziente-specifica per decidere se l'intervento terapeutico debba limitare le compensazioni e cercare di migliorare la menomazione, oppure procedere con l'allenamento dei modelli compensatori con l'obiettivo di migliorare la sicurezza dei movimenti, sostenere l'indipendenza funzionale nelle attività della vita quotidiana e promuovere la partecipazione sociale [87].

Diversi argomenti sostengono l'importanza di puntare al miglioramento funzionale attraverso la riduzione della menomazione. In primo luogo, sebbene i movimenti compensatori possano aiutare i pazienti a svolgere compiti a breve termine, la presenza di compensazioni può essere associata a problemi a lungo termine, come la riduzione della mobilità articolare e il dolore. Recenti ricerche sulla capacità di plasticità neuronale suggeriscono che, con un allenamento adeguato, i miglioramenti motori dell'arto superiore possono continuare anche nella fase cronica dell'ictus. [79]. In secondo luogo, consentire l'uso di compensazioni motorie potrebbe portare a un modello di *learned non-use*, limitando la capacità di successivi miglioramenti nella funzione motoria del braccio [79]. A sostegno di questa teoria, nella *review* (revisione sistematica) su RCT (*Randomized Controlled Trial*) condotta da Pain et al. (2015) [88] diversi studi hanno riportato miglioramenti nella funzione e riduzione dei movimenti compensatori a seguito di esercizi di *reaching* limitando i movimenti del tronco *(trunk restraint*). Tuttavia, i pazienti inclusi negli studi sono stati sottoposti a criteri di inclusione, suggerendo che questa tecnica è adatta a pazienti con una certa abilità [88].

Nel contesto della teleriabilitazione asincrona, dove il terapista non è fisicamente presente per imporre restrizioni o fornire feedback correttivi manuali, il rischio che il paziente consolidi *pattern* di *learned bad-use* è elevato. Pertanto, risulta opportuno implementare sistemi tecnologici in grado di rilevare automaticamente i movimenti compensatori in modo da offrire una panoramica più completa sulla qualità degli esercizi svolti ed eventualmente fornire un biofeedback per correggere questi schemi motori non fisiologici.

Rilevazione, valutazione e correzione dei movimenti compensatori

La quantificazione oggettiva delle strategie compensatorie si avvale di diverse soluzioni tecnologiche. Una recente *review* condotta da Wang et al. (2022) [89] ha classificato gli approcci esistenti identificando tre categorie tecnologiche principali utilizzate per il monitoraggio dell'arto superiore post-ictus:

- Sistemi *marker-based* (sistemi optoelettronici), rappresentano il *gold standard* per la loro elevata accuratezza nel tracciamento cinematico in tempo reale e vengono frequentemente impiegati per validare l'efficacia di altri sensori. Tuttavia, l'adozione clinica massiva di questi sistemi è limitata da costi elevati, complessità di *setup* che richiede personale esperto e dalla necessità di operare in ambienti controllati e strutturati. Questo approccio non è applicabile in teleriabilitazione.
- Sensori indossabili: tecnologia più diffusa, circa il 35% degli studi della review, composti da unità di misura inerziali (IMU) posizionati su segmenti corporei chiave come tronco, spalla e gomito. Rispetto ai sistemi ottici, i sensori indossabili offrono vantaggi significativi in termini di portabilità, costi contenuti e semplicità di configurazione, rendendoli particolarmente idonei per il monitoraggio in ambienti meno supervisionati come il domicilio. Tuttavia, lo svantaggio principale risiede nel rischio che il dispositivo stesso, essendo attaccato al corpo, possa indurre movimenti innaturali. Soluzioni miniaturizzate sono ormai uno standard commerciale.
- Sensori ottici marker-less come Microsoft Kinect o videocamere RGB, utilizzati nel 22% dei casi, offrono un compromesso tra usabilità e funzionalità, eliminando la necessità di applicare dispositivi sul corpo del paziente e riducendo i tempi di preparazione. Sebbene siano ideali per l'uso domestico grazie alla loro natura non invasiva, presentano limitazioni legate alla precisione delle misure e potenziali problematiche relative alla privacy del paziente.

Per rilevare la compensazione, l'approccio più diffuso consiste nell'impostare delle soglie sugli angoli o sullo spostamento dei segmenti corporei coinvolti nel movimento. Queste soglie possono essere fisse, oppure personalizzate. Le soglie fisse utilizzano valori predefiniti, derivati principalmente dall'analisi del movimento dei soggetti sani, offrendo semplicità e riproducibilità, ma limitati dalla scarsa adattabilità a diversi gradi di disabilità. In questo contesto Duff et al. (2010) [90] effettuano un'analisi del *reaching* tramite un sistema *marker-based* e telecamere a infrarossi, individuando 3 compensazioni: elevazione della spalla (relazione angolare della proiezione del *marker* della spalla sul piano del busto), protrazione della spalla (misurata dalla distanza dal *marker* della spalla al piano del busto) , flessione e rotazione del busto, analizzando la differenza con i valori ottenuti dai soggetti sani. Valdés et al. (2018) [91] utilizzando un dispositivo Kinect v2, definiscono una soglia di 3 cm sul movimento in direzione anteriore del giunto spalla-colonna vertebrale, gli spostamenti inferiori a questo valore sono considerati compensazioni fisiologiche e non attivano il feedback. Luo et al. (2023) [92] hanno determinato soglie specifiche confrontando le deviazioni massime dei soggetti sani durante esercizi di flessione della spalla e abduzione orizzontale. Le soglie raccomandate sono: 1,00 cm per la

flessione laterale o estensione del tronco, 2,00-3,00 cm per l'elevazione della spalla e 1,00-2,00 cm per la depressione della spalla, a seconda dell'esercizio specifico. Passon et al. (2020) [93] identificano i movimenti compensatori se lo spostamento della spalla supera i ±10 cm per un tempo superiore al 20% della durata del ciclo di movimento.

Le soglie personalizzate si calibrano sul paziente, garantendo adattabilità e progressività ma dipendenti dalla qualità della calibrazione. Foreman e Engsberg (2019) [94] utilizzano un algoritmo di calibrazione che misura la flessione media del tronco durante movimenti non vincolati. La soglia per il feedback viene quindi impostata al 90% di questo valore medio iniziale per incoraggiare il paziente a ridurre la compensazione. Chen et al. (2016) [95] definiscono una soglia ± 5 gradi per il tronco e ± 10 gradi per la spalla rispetto alla variazione angolare misurata nel *reaching* guidato dall'esaminatore. Nguyen et al. (2021) [96] propongono una rilevazione della compensazione del tronco tramite PTOR (*Percent Time Out of Range*). Gli angoli del tronco sono misurati rispetto alla posizione in piedi, tramite un singolo IMU posizionato sullo sterno. Per ciascun compito e per ciascun asse, si definisce una "zona nominale" di movimento del tronco, calcolata durante i movimenti con l'arto sano, come intervallo media ± 2σ (σ: deviazione standard). Il PTOR è definito come la percentuale di tempo in cui l'angolo del tronco cade al di fuori di questo intervallo. Gli autori osservano che valori di PTOR superiori a circa il 30% risultano associati a movimenti compensatori chiaramente apprezzabili clinicamente e propongono tale valore come soglia indicativa.

Parallelamente agli approcci basati su soglia, in letteratura sono stati implementati algoritmi di *Machine Learning* (ML) e *Deep Learning* (DL) per l'analisi di schemi di movimento complessi per rilevare compensazioni.

Un esempio di apprendimento supervisionato (ML) applicato a dati cinematici è stato proposto da Fu et al. (2020) [97] e prevede l'uso di due telecamere RGB, per estrarre gli angoli articolari. Due classificatori *multi-label*, *Multi-Label k-Nearest Neighbor (ML-KNN)* e *Multi-Label Decision Tree (ML-DT)* sono stati addestrati *frame per frame* utilizzando, per ciascun istante temporale, gli angoli di flessione e rotazione del tronco, gli angoli di flessione e abduzione della spalla e l'angolo di estensione del gomito. I classificatori hanno ottenuto buone prestazioni nel rilevamento dei compensi del tronco, ma non quelli della spalla. Khoramdel, et al. (2021) [98] utilizzano metodi di *Deep Learning* che analizzano le coordinate articolari nel tempo. Tra i vari modelli testati su dati catturati tramite Kinect, l'architettura *Transformer* a due strati e *Gated Recurrent Units (GRU)* si sono rivelate più performanti, rispetto alle *Recurrent Neural Networks (RNN)* e al *Long Short-Term Memory (LSTM)*. Esistono inoltre metodi che puntano alla riduzione della dimensionalità dei dati per identificare le compensazioni, come l'Analisi delle Componenti Principali (PCA) proposta da Nordin et al. (2016) [99]. In questo scenario, la PCA viene applicata alle posizioni 3D dei 7 giunti del tronco acquisiti con Kinect v2, creando un modello del torace. Tale modello è composto da 3 vettori con 3 dimensioni, riducendo le *features* da 18 a 9. La numerosità delle *features* può essere ridotta ulteriormente considerando solo la componente lungo Z del modello, oppure le differenze tra i vettori o angoli PCA in *frame* successivi (3 *features*) e indici riassuntivi (differenza media in valora assoluto, angolo medio - 1 *feature*). Le sequenze di movimento (80 *frame* per task) vengono poi usate per addestrare vari classificatori (Naïve Bayes,

SVM, KNN, Random Forest, ecc.) per distinguere movimenti normali da movimenti con compenso indotti tramite tutore al gomito. Gli autori hanno ottenuto accuratezze molto elevate, superiori al 90%, specialmente con l'SVM sul solo terzo componente principale. Ranganathan et al. (2017) [100] presentano un metodo di classificazione estraendo semplici *features* statistiche, come la deviazione standard dell'accelerazione in direzione anteroposteriore del tronco e la velocità angolare di flesso-estensione del gomito, ottenute tramite 3 IMU posizionati su sterno, braccio e avambraccio. Queste due *features* hanno ottenuto risultati più accurati nel rilevare i compensi del tronco, durante attività di *reaching* verticale e orizzontale rispetto agli approcci con 6 features (accelerazioni e angoli dell'IMU sul tronco), o 18 features (accelerazioni e angoli di tutti gli IMU).

Oltre ai classici sistemi di *motion capture,* l'elettromiografia di superficie (sEMG) si è dimostrata uno strumento efficace per identificare *pattern* di attivazione muscolare anomala associata ai movimenti compensatori. Ma et al. (2019) [101] hanno sviluppato un metodo di rilevazione del compenso del tronco basato esclusivamente su sEMG, analizzando l'attività di nove muscoli, tra cui retto addominale, obliqui esterni ed erettori spinali. Attraverso l'estrazione di feature nel dominio del tempo come *Root Mean Square* (RMS) e *Mean Absolute Value* (MAV) e l'uso di classificatori SVM, lo studio ha dimostrato la fattibilità di rilevare i compensi del tronco (flessione e rotazione) e l'elevazione della spalla non solo in soggetti sani, ma anche in pazienti post-ictus, sebbene con prestazioni leggermente inferiori dovute alla variabilità dei *pattern* patologici. Xu et al. (2022) [102] hanno proposto una *trunk restraint belt* che combina sensori di forza, spostamento angolare e sEMG per monitorare le compensazioni durante il *reaching*. In particolare, il segnale sEMG prelevato dal muscolo trapezio superiore si è rivelato un indicatore fondamentale per distinguere il compenso di elevazione della scapola.

Per la correzione dei movimenti compensatori durante il *reaching* vengono utilizzate diverse tipologie di feedback: uditivi, visivi, aptici o una combinazione di essi. Il feedback uditivo utilizza segnali sonori per richiamare l'attenzione del paziente quando viene rilevata una compensazione. Questi possono consistere in allarmi [94], suoni dissonanti [90], [95] o emotivamente negativi [103] e indicazioni verbali che suggeriscono esplicitamente come correggere il movimento [104]. Per quanto concerne il feedback visivo, possono essere adottate varie strategie:

- Informare il paziente: utilizzo di avatar che riproducono il movimento, elementi grafici (come barre, cerchi o linee) che rappresentano i limiti di spostamento consentiti, codici colore (tipicamente verde per l'esecuzione corretta, rosso per l'errore) o testi informativi.[91], [92], [94], [103], [104].
- Penalizzare il movimento: annullamento della ripetizione [94], aumento della difficoltà del task (ad esempio spostando il target proporzionalmente al movimento del tronco [105]), introduzione di elementi penalizzanti nel gioco riabilitativo (ad esempio la perdita di punti o ostacoli [103]).
- Ricompensare i movimenti corretti o ridurre la difficoltà: riduzione automatica del Range of Motion (ROM) richiesto per completare la ripetizione al fine di limitare la necessità di compensare, introduzione di ricompense o punti bonus nel gioco [92].

Un'ulteriore feedback che può essere fornito è quello di forza (aptico), che può essere di natura assistiva, nel quale il movimento viene agevolato [104], oppure penalizzante, costringendo il soggetto a esercitare una forza maggiore qualora vengano rilevati movimenti compensatori [91].

Oltre a queste modalità, l'uso del biofeedback elettromiografico (sEMG-BFB) rappresenta una strategia promettente, sebbene meno esplorata specificamente per la riduzione dei compensi. Lirio-Romero et al. (2020) [106] tramite uno studio RCT hanno dimostrato l'efficacia di un intervento di 6 settimane basato su sEMG-BFB focalizzato sul controllo motorio della scapola. In questo studio, il biofeedback non è stato utilizzato solo per aumentare l'attivazione dei muscoli agonisti, ma specificamente per ridurre l'attività del trapezio superiore, responsabile dell'elevazione della spalla. I risultati hanno evidenziato che il gruppo sottoposto a sEMG-BFB ha ottenuto miglioramenti significativi nella funzione motoria dell'arto superiore e nel ROM attivo, riducendo contestualmente l'attivazione del trapezio superiore a favore di un reclutamento più fisiologico.

Tuttavia, nonostante l'elettromiografia sia ampiamente utilizzata per migliorare la funzione dell'arto superiore o per analizzare le sinergie muscolari post-ictus [107], la letteratura attuale presenta pochi studi che impiegano il biofeedback sEMG con l'obiettivo esplicito e primario di ridurre i movimenti compensatori, preferendo approcci cinematici (IMU o sensori ottici) per questo scopo.

Una parte del presente lavoro di tesi, si inserisce in questo contesto, proponendo un sistema di rilevazione dei compensi di abduzione della spalla durante il *reaching*, tramite analisi del segnale sEMG del deltoide laterale.

Materiali e metodi

Architettura del sistema

In Figura 11 è riportata una rappresentazione schematica dell'apparato sperimentale necessario per il *training* riabilitativo e i relativi componenti hardware e software coinvolti. Il setup è composto da un *headset* VR per la visualizzazione e l'interazione con l'ambiente virtuale e uno o più dispositivi per l'acquisizione del segnale elettromiografico di superficie (sEMG). Quest'ultimi sono interfacciati tramite Wi-Fi a un PC dedicato, il quale gestisce l'acquisizione, l'elaborazione in tempo reale dei dati e il successivo invio all'applicazione VR. Nello specifico, per questo studio sono stati impiegati il visore Meta Quest 3 e i sensori Due+ prodotti da OT Bioelettronica. La Figura 11 mostra l'impiego di una maglia sensorizzata, un prototipo progettato e sviluppato dal laboratorio MeDSP (*Medical Devices and Signal Processing*) dell'Università di Cagliari. Tale indumento è stato funzionalizzato bilateralmente con elettrodi tessili, realizzati mediante tecniche di *spray coating* e *screen printing* su un tessuto elastico in poliestere. In questa configurazione, il sensore è assicurato alla maglia tramite un sistema di fissaggio in velcro; in alternativa, il sistema prevede la possibilità di utilizzare elettrodi gelificati standard e le fasce di fissaggio fornite dal produttore dei sensori.

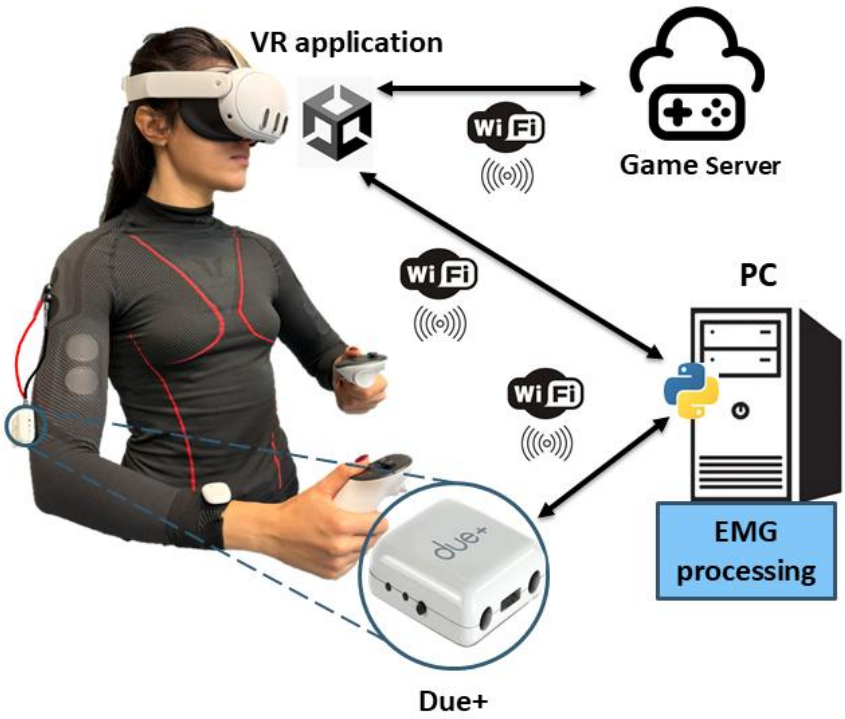

Figura 11: Architettura del sistema proposto: l'applicazione VR sviluppata in Unity è installata sul visore che comunica tramite Wi-Fi sia al PC che al Game Server. La sonda Wi-Fi EMG è collegata al PC, il quale gestisce l'acquisizione e l'elaborazione dei dati, successivamente inviati all'applicazione.

L'applicazione Unity comunica tramite protocollo Wi-Fi con un Game Server, elemento cardine dell'architettura *multiplayer*. In questo scenario, il server funge da autorità centrale preposta alla gestione e alla sincronizzazione dello stato del gioco tra tutti i client connessi. La sua funzione principale è garantire che l'ambiente virtuale sia coerente per ogni partecipante,

aggiornando in tempo reale le posizioni dei remi e l'avanzamento della barca. Oltre alla sincronizzazione delle variabili fisiche, il server si occupa della coordinazione logistica della sessione. Esso organizza gli utenti nelle diverse lobby in base alla lateralità selezionata, in modo da riempire correttamente la barca virtuale composta da 8 rematori (4 destri e 4 sinistri). All'interno dell'ambiente *multiplayer* ad ogni partecipante viene assegnata una posizione fissa nella barca virtuale, rispettando l'alternanza tra remo destro e sinistro, tuttavia, per ottimizzare l'esperienza collaborativa, la prospettiva locale viene traslata all'ultima postazione. Questa discrepanza tra posizione logica (sul server) e posizione locale (sul client) garantisce che ogni utente abbia una visuale libera di tutti gli altri rematori (Figura 12).

Figura 12: Visuale locale (lato client) degli utenti nella barca virtuale.

Nell'eventualità in cui non siano presenti un numero sufficiente di giocatori nella lobby, dopo un certo intervallo temporale, il server avvia l'*exergame* istanziando un numero sufficiente di *bot player* nella barca per riempire i posti vacanti. Questi *Virtual Agents* (VA) sono completamente indistinguibili nel mondo virtuale dai giocatori reali e hanno l'obbiettivo di svolgere l'esercizio con gli utenti implementando un comportamento *human-like* capace di adattarsi alle performance dell'equipaggio. L'applicazione prevede una *spectator mode* per il terapista che gli permette di accedere alla sessione in corso e monitorarla da remoto.

Integrazione della strumentazione in ambiente Unity 3D

In questa sezione vengono descritte le caratteristiche e l' integrazione nella piattaforma Unity dei dispositivi che compongono il setup sperimentale: il visore Meta Quest 3, utilizzato per la visualizzazione dello scenario VR, con i relativi controller, impiegati per il tracciamento della

posizione delle mani e l'interazione con l'ambiente virtuale e le sonde Due+ per l'acquisizione del segnale sEMG.

Caratteristiche dei sensori

Meta Quest 3

Il Meta Quest 3 (Figura 13) è un dispositivo commerciale *stand-alone*, che integra in un unico sistema capacità di calcolo, display e sistema di tracciamento, senza necessità di utilizzare dispositivi esterni.

Figura 13: Visore Meta Quest 3 e controller Meta Quest Touch Plus

Il visore impiega due pannelli LCD con ottiche a lenti *pancake*, con una risoluzione di circa 2064 × 2208 pixel per occhio, *refresh rate* configurabile a 72, 90 o 120 Hz e un campo visivo di 110 gradi in orizzontale e 96 gradi in verticale [108]. Il dispositivo adotta una tecnologia di tracciamento *inside-out*, basato su una combinazione di sensori inerziali e di sei fotocamere, 4 IR e 2 RGB, che mappano l'ambiente circostante a sei gradi di libertà (DoF). L'interazione con l'ambiente può avvenire in primo luogo attraverso il tracciamento delle mani (*hand tracking*), che utilizza le telecamere del visore per catturare la posizione e il movimento delle dita in tempo reale. Questi dati sono elaborati tramite algoritmi di *computer vision* e modelli di *machine learning* che riconoscono pose e gesti specifici, interpretandoli come comandi. Il sistema opera in diverse modalità a seconda delle necessità dell'applicazione: una modalità predefinita a 30 Hz per un bilanciamento tra precisione e consumo, una modalità per movimenti rapidi (*Fast Motion Mode*) a 60 Hz per ridurre la latenza in scenari dinamici, e una modalità a movimento ampio (*Wide Motion Mode*) che, tramite la tecnologia *Inside Out Body Tracking* (IOBT), stima la posizione delle mani anche quando queste escono temporaneamente dal campo visivo delle telecamere. Il sistema gestisce inoltre l'accuratezza attraverso una calibrazione automatica che misura la dimensione della mano dell'utente, assicurando una rappresentazione virtuale coerente. È inoltre prevista una modalità multimodale che permette l'uso simultaneo di mani e controller per scenari di interazione complessi [108], [109].

I controller Meta Quest Touch Plus sono anch'essi tracciati in 6 DoF grazie a una combinazione di sensori interni e il rilevamento da parte del visore. Ogni controller integra un proprio IMU (accelerometro e giroscopio) e una serie di LED a infrarossi. Il tracciamento del controller può garantire una precisione inferiore al millimetro, grazie all'uso sia della visione artificiale classica che dei modelli di apprendimento automatico. Le telecamere del visore identificano e triangolano i LED a infrarossi sul controller fornendo una precisione inferiore al pixel, mentre i modelli di apprendimento automatico sono addestrati per stimare la posizione del controller quando i LED sono occlusi e in condizioni di illuminazione difficili. L'IMU può essere utilizzato per stimare la posizione e l'orientamento per un breve periodo di tempo quando il controller è fuori dal campo visivo delle telecamere [109].

Infine, sebbene il Meta Quest 3 non integri sensori interni per il tracciamento oculare *(eye-tracking*), esso implementa un sistema di *gaze detection* basato sull'orientamento della testa (*head-gaze*), dove il centro del campo visivo del visore funge da puntatore per l'interazione con gli elementi dell'interfaccia. Per quanto concerne il riconoscimento vocale, il dispositivo integra una serie di microfoni e supporta nativamente il *voice recognition* tramite il Voice SDK di Horizon OS, permettendo all'utente di impartire comandi vocali o interagire con le interfacce utente [109].

Sonde Due+ e SyncStation Bipolar

I Due+ di OT Bioelettronica sono dispositivi indossabili e wireless che permettono di acquisire fino a 2 segnali sEMG. La sonda integra anche un'unità di misura magneto-inerziale (MIMU) composta da accelerometro, giroscopio e magnetometro, configurata in modalità Fusion Mode - NDOF, dalla quale vengono derivati quattro quaternioni utilizzati per descrivere l'orientamento del dispositivo. I segnali sEMG acquisiti dalla sonda Due+ sono registrati in modalità bipolare. Oltre ai due canali di ingresso per gli elettrodi bipolari (CDE-C o CDE-S) è presente un terzo ingresso al quale è possibile collegare un elettrodo di riferimento. Per la raccolta dei dati EMG, viene implementato un filtro passa-alto firmware con una frequenza di taglio di 10.5 Hz, durante il campionamento dei segnali a 2000 Hz, che rimuove la componente DC e sposta la linea di base dei segnali al centro della dinamica. Oltre ai 2 segnali biologici e i 4 canali provenienti dal sensore inerziale, sono presenti anche 2 canali accessori. La risoluzione reale dei dati del quaternione è pari a 14 bit, estesa con segno a 16 bit in funzione della modalità di acquisizione della sonda Due+. I quaternioni sono frutto di un calcolo interno al sensore inerziale e sono aggiornati ad una frequenza di 100 Hz, quindi, nel caso di campionamento a 2000 Hz, ci saranno 20 campioni con gli stessi valori dei quaternioni prima di ottenere un nuovo set di valori. I due canali accessori contengono informazioni relative al segnale di sincronizzazione inviato dalla SyncStationBipolar, all'utilizzo del *buffer* di memoria interno alla sonda Due+ ed un contatore dei campioni [110]. In Figura 14 è mostrato l'orientamento del sensore inerziale.

La sonda Due+ può essere usata singolarmente, in modalità *AccessPoint* con una connessione diretta con il PC. Questa modalità permette di acquisire esclusivamente i segnali provenienti da una sonda Due+ senza possibilità di sincronizzazione con altre sonde o altri dispositivi (in questa modalità il sistema è chiamato Due+Lite). La sonda Due+ funge da server DHCP al quale può collegarsi un solo dispositivo alla volta. Sebbene la rete sia generata dalla sonda Due+, è il PC (o

tablet, smartphone ecc..) a dover aprire un TCP *socket* con il ruolo di server a cui la sonda Due+ cercherà di connettersi. La sonda infatti conosce l'indirizzo IP del PC, in quanto è stato assegnato dalla sonda stessa con il protocollo DHCP. L'indirizzo IP della sonda Due+ è 192.168.14.ID, dove ID rappresenta il numero identificativo della sonda. Una volta stabilita la connessione TCP è possibile inviare il byte di controllo alla sonda per configurare l'acquisizione e dare il via al trasferimento dati. Questi ultimi vengono bufferizzati all'interno della sonda e sono inviati in pacchetti di circa 1400 byte appena possibile. Se il PC non è in grado di ricevere dati o se la connessione Wi-Fi non lo permette, i dati iniziano ad accumularsi all'interno dei *buffer*. Quando il *buffer* si riempie completamente, i nuovi dati verranno persi fino a quando il PC e la connessione Wi-Fi non saranno nuovamente disponibili per il trasferimento [110].

Quando si sincronizzano più sonde tramite la SyncStation Bipolar/Bipolar+ (SSB) (Figura 14), il sistema è chiamato Due+Pro. La SSB è un'unità che funge simultaneamente da stazione di ricarica, modulo di comunicazione e sincronizzazione per un massimo di otto sonde. Il dispositivo è progettato per aggregare, sincronizzare e inoltrare tutti i dati provenienti dalle sonde e dagli eventuali ingressi ausiliari verso un unico punto di acquisizione sul PC. La SSB genera una rete Wi-Fi dedicata e non visibile all'utente alla quale le sonde si collegno automaticamente, a meno che non siano impostate in modalità *AccessPoint.* Il PC si connetta al dispositivo tramite cavo Ethernet e invia una stringa di controllo per configurare e avviare la trasmissione dei dati [110].

Figura 14: Sistema Due+Pro. SyncStationBipolar (a sinistra) e singola sonda Due+ (a destra) con relativo orientamento del sensore inerziale IMU. Figura tratta da [125].

Per ogni dispositivo connesso e per gli ingressi ausiliari viene allocata una certa quantità di memoria come *buffer* di ingresso, tale da contenere 32 pacchetti Wi-Fi per ogni sonda. I dati ricevuti vengono temporaneamente scritti nei rispettivi *buffer* in attesa di essere processati e copiati in un *buffer* di uscita. Se i dati provenienti da alcuni dispositivi non vengono ricevuti entro un limite di tempo massimo, il pacchetto di output per il PC viene comunque inviato inserendo

valori pari a 0 nei campioni del dispositivo mancante. Il *buffer* di uscita può contenere fino a un totale di 210 pacchetti. Il limite massimo per l'invio di dati al PC, anche in assenza di campioni da tutte le sonde, è di *default* 200 pacchetti accumulati nel *buffer*. Questo parametro è modificabile e prende il nome di latenza. Una volta assemblato il pacchetto, la SSB lo invia al PC attraverso la porta Ethernet mediante un *socket* TCP. Il flusso risultante appare al PC come se provenisse da un unico dispositivo, con un numero totale di canali pari alla somma dei canali di tutte le sonde collegate più 6 canali ausiliari. Per le sonde Due+ la dimensione del pacchetto Wi-Fi è 864 byte, corrispondente a 54 campioni, inviati ogni 27 millisecondi [110].

Nell'applicazione sviluppata, l'utilizzo della sonda singola permette esclusivamente l'acquisizione del segnale elettromiografico del deltoide laterale, utilizzato come biofeedback durante lo svolgimento dell'*exergame.* L'impiego della SSB invece, permette anche l'acquisizione di segnali provenienti da altri muscoli (ad esempio bicipite e tricipite), destinati esclusivamente a scopi di monitoraggio e analisi post-sessione.

Implementazione software

Integrazione Meta Quest 3

L'integrazione del Meta Quest 3 su Unity è avvenuta mediante l'utilizzo dell'asset *XRInteractioToolkit*. Nell'applicazione è stato creato un Game Object[1] (elemento di gioco, GO) denominato *VRPlayer*, che contiene al suo interno tutti gli elementi e le classi per la gestione della presenza dell'utente e delle sue capacità di interazione con l'ambiente. Il GO contiene l'*XR Origin*, il quale ha il compito di sincronizzare gli spostamenti del visore e dei controller nel mondo reale all'interno della scena virtuale. *L'XR Origin* contiene nella sua gerarchia due GO fondamentali:

- *Main Camera*: rappresenta il punto di vista del giocatore nella scena Unity ed è vincolata ai movimenti del visore.
- Controller destro e sinistro: la cui posizione e l'orientamento sono sincronizzati con quelli dei controller del mondo reale. Questo è reso possibile dal componente *XR Controller* (*Device-based*) che riceve i dati di tracciamento direttamente dall'hardware. I componenti *XR Controller* sono stati configurati con la modalità di *Update Tracking Type* impostata su *Update AND Before Rende*r. Questa impostazione garantisce che la posizione e l'orientamento dei controller vengano aggiornati sia durante il ciclo logico che immediatamente prima del *rendering* della Scena.

Gli elementi del mondo virtuale con cui l'utente può interagire, definiti *interactables* (come i componenti dell'interfaccia utente o oggetti di gioco), e i dispositivi di input dell'utente, denominati *interactors*, non comunicano direttamente. La loro interazione è mediata dall'*XR Interaction Manager*, un componente che agisce come elemento centrale per l'intero sistema di interazione. Questo modulo opera come un supervisore logico che assicura la sincronizzazione e la coerenza degli stati logici degli oggetti durante le interazioni all'interno dell'ambiente virtuale, prevenendo conflitti e garantendo che ogni azione avvenga secondo una sequenza prestabilita. Il ciclo di interazione si articola in tre stati sequenziali. Il processo inizia con lo stato di *Hover*,

[1] Un Game Object in Unity è l'entità fondamentale che ospita dei Componenti (script, mesh, collider, rigidbody ecc) per definire comportamento, aspetto e fisica nell'applicazione.

attivato dal semplice puntamento o dalla prossimità, che fornisce feedback visivi (come il cambiamento di colore di un pulsante). Segue lo stato di *Select*, innescato da un'azione esplicita dell'utente come la pressione di un trigger, che formalizza il legame logico tra *interactor* e *interactable*. Infine, lo stato di *Activate* permette di eseguire funzioni secondarie sull'elemento già selezionato.

La gestione dell'input hardware è affidata *all'Input Action Manager*, il quale funge da ponte tra i segnali fisici dei controller e la logica dell'applicazione. Attraverso l'impiego degli *Input Action Assets*, le interazioni svolte dall'utente sul controller (come la pressione di un tasto o il movimento di una levetta analogica) vengono mappate su azioni logiche nella Scena Unity. Nell'applicazione sviluppata è stato utilizzato l'*Input Action Assets* predefinito fornito dall'*XR Interaction Toolkit* il quale garantisce la compatibilità con diverse piattaforme hardware per la realtà virtuale. In particolare, per il Meta Quest 3 il tasto *Grip* è associato all'azione di *Select* per afferrare oggetti, il *Trigger* (grilletto) è riservato allo stato di *Activate* o al click sulle interfacce,

Per l'interazione a distanza con le interfacce utente (UI) ad ogni GO relativo ai controller è stato associato l'*XR Ray Interactor*. Questo componente implementa un sistema basato su *ray-casting* che proietta un raggio virtuale nello spazio, con origine il GO e direzione *forward* (in avanti/asse z), e verifica se quest'ultimo interseca eventuali oggetti nella Scena. Il raggio è reso visibile all'utente attraverso l'integrazione di un *Line Renderer* e dell'*XR Interactor Line Visual*, i quali generano un raggio luminoso, che facilita il puntamento e cambia colore quando interseca un oggetto, passando da rosso a bianco. Per rendere le UI interattive all'interno dell'ambiente virtuale è indispensabile l'integrazione *dell'XR UI Input Module* nell'*Event System* di Unity. Questo componente è necessario poiché le interfacce utente (UI) sono progettate per interpretare input bidimensionali, come quelli di un mouse o di un touch screen. *L'XR UI Input Module* funge da interprete traducendo i dati provenienti dal controller o dai *ray-cast* in eventi di puntamento compatibili con il sistema di coordinate delle UI. Senza questo modulo, il sistema non sarebbe in grado di convertire la collisione del raggio virtuale nello spazio 3D in azioni specifiche dell'interfaccia, come il "click" o l'invio di un comando. In questo modo, l'utente è in grado di interagire con gli elementi presenti nelle UI, posizionate nell'ambiente virtuale.

Modalità operative previste dal sistema di teleriabilitazione

L'applicazione è stata progettata per operare secondo due macro-modalità di esecuzione, che si differenziano in base al dispositivo su cui è installata l'applicazione e al protocollo di comunicazione per la ricezione dei dati EMG:

1. **Modalità PC-Link** (Figura 15): In questa configurazione, l'applicazione risiede ed è eseguita sul computer. Il sistema VR viene utilizzato come periferica di output visivo e input di tracciamento, collegandosi al PC tramite tecnologia *Meta Quest Link* (via cavo o Air Link). In questo scenario, il sistema di acquisizione è connesso direttamente al PC, che gestisce sia l'esecuzione dell'applicazione che l'acquisizione, l'elaborazione e la visualizzazione dei segnali. Questa modalità supporta due varianti di connessione hardware:

   - Connessione alla SyncStation**:** avviene tramite protocollo TCP sulla porta 54320**.**
   - Connessione alla sonda Due+: avviene tramite protocollo TCP sulla porta 54321.

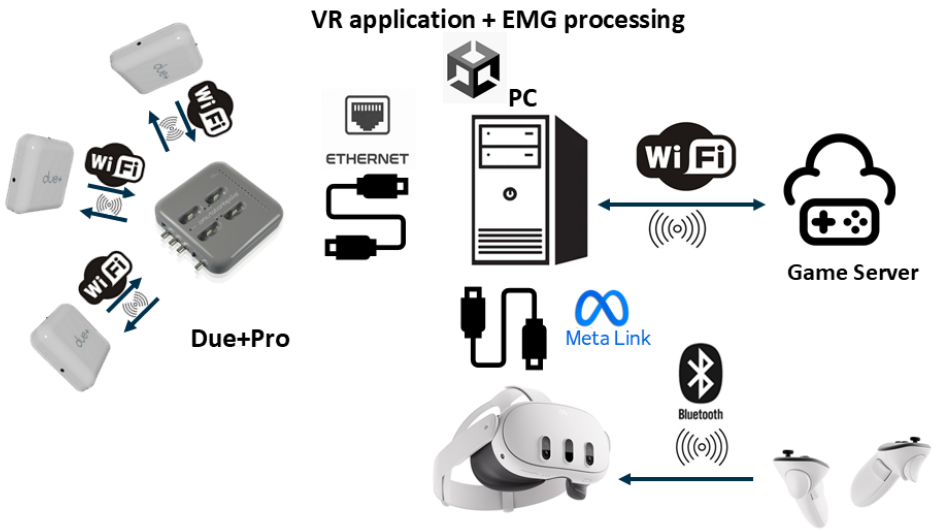

Figura 15: Schema delle connessioni nella modalità PC-Link nel caso di utilizzo della SyncStationBipolar (sistema Due+pro)

2. **Modalità *stand-alone*** (Figura 16): In questa modalità, l'applicazione è installata e viene eseguita direttamente sull'hardware del visore. La sonda non può essere collegata direttamente al Meta Quest, in quanto il canale Wi-Fi è utilizzato per la connessione al Game Server. Per questo motivo, viene utilizzato un computer come ponte tra il sistema di acquisizione e il visore. La sonda Due+ o la SyncStation, sono collegate al PC, il quale riceve i dati grezzi, ne effettua l'elaborazione e trasmette al visore, tramite protocollo UDP, esclusivamente il dato processato del segnale elettromiografico.

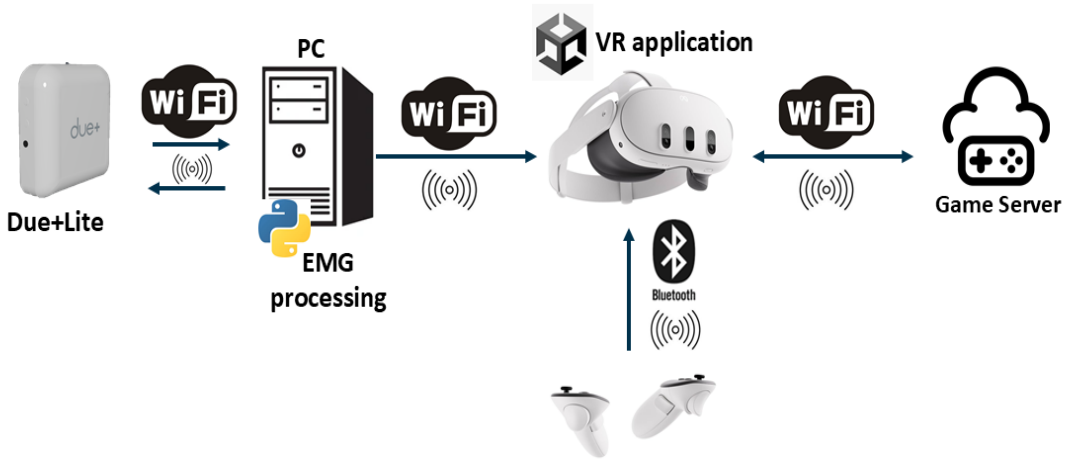

Figura 16: Schema delle connessioni nella modalità stand-alone nel caso di utilizzo della sonda singola (sistema due+Lite)

Nel caso venga utilizzata una singola sonda, in entrambe le modalità, è necessario impiegare un dispositivo di estensione della rete Wi-Fi. Ciò consente di separare il traffico di rete in due canali distinti: uno dedicato esclusivamente alla comunicazione con la sonda, l'altro destinato alla connessione con il Game Server (modalità PC-Link) o alla comunicazione con il visore (modalità *stand-alone*).

Connessione e comunicazione con le sonde

La comunicazione e la gestione dei sensori EMG Wi-Fi è stata integrata in Unity mediante uno script dedicato (*SyncStationConnection.cs*) utilizzando il framework .NET. La classe coordina la connessione TCP/UDP, il *parsing* dei pacchetti e la memorizzazione dei dati acquisiti in *buffer* dedicati. All'avvio, lo script crea un *thread* secondario dedicato alla connessione che esegue il metodo *custom ConnectToSyncStation()*, evitando di bloccare il *main thread* (flusso si elaborazione principale) di Unity. L'obiettivo di tale funzione è stabilire una connessione in una delle due modalità operative senza impostarne una a priori. Per la modalità PC-Link il sistema tenta di connettersi al server TCP generato dalla SSB (IP: 192.168.76.1, porta: 54320) tramite il metodo *BeginConnect* della classe *TcpClient* (libreria *System.Net.Sockets*), il quale avvia il tentativo di *handshake* TCP su un *thread* separato gestito separatamente dal framework .NET. Il metodo restituisce immediatamente il controllo al *thread* chiamante tramite un oggetto che implementa l'interfaccia *IAsyncResult*, permettendo di gestire l'operazione asincrona.

Attraverso il metodo *AsyncWaitHandle.WaitOne* dell'interfaccia, il sistema sospende l'esecuzione del *thread* avviato da *BeginConnect* dopo un tempo prestabilito (5 s). Una volta stabilita la connessione, il sistema aggiorna gli indicatori di stato interni e inizializza i contenitori dei dati in base al numero di dispositivi abilitati. Contestualmente, viene allocato lo *stream* TCP tramite il metodo *GetStream*() della classe *TcpClient*. Questo oggetto, di tipo *NetworkStream*, rappresenta il canale di comunicazione nel quale transitano i dati dei sensori, mettendo a disposizione metodi nativi per il monitoraggio e la lettura del flusso di informazioni.

In caso di fallimento o superati i 5 s, il sistema registra l'esito e prosegue automaticamente tentando la connessione con la sonda singola. Per la connessione con la sonda singola il metodo crea un server TCP locale sul PC, tramite la classe *TCPListener di* .NET*,* a qualsiasi indirizzo IP e porta 54321. Tale approccio è stato adottato per permettere al sistema di gestire la connessione indipendentemente dall'indirizzo IP assegnato alla sonda. Poiché il metodo di accettazione dei client della libreria standard, ovvero *TCPListener.AcceptTcpClient*(), non presenta nativamente un meccanismo di *timeout*, è stato implementato un sistema *custom* basato su *polling* temporizzato: il server viene interrogato ciclicamente (ogni 50 ms) per verificare la presenza di connessioni in ingresso, attraverso il metodo *TCPListener.Pending()*, mentre l'utilizzo di un cronometro *built-in* (*Stopwatch*) monitora il tempo trascorso. Se il metodo *Pending* rileva un tentativo di connessione, il sistema accetta il *TcpClient e* inizializza lo *stream* TCP, in maniera analoga alla modalità precedente.

Qualora non venga rilevato alcun client connesso al TCP *socket* dopo il tempo limite di 5 s il sistema procede con l'ultima modalità operativa effettuando un tentativo di connessione ad un PC server. In questo caso, la connessione avviene tramite UDP cioè un protocollo orientato al datagramma e non alla connessione. A differenza di TCP, che richiede un *handshake* esplicito, l'UDP non stabilisce alcuna sessione tra client e server. Per verificare l'esistenza di una connessione è stato creato un metodo dedicato (*CheckRemoteConnection*()), il quale inizializza la comunicazione avvalendosi della classe *UdpClient* (libreria *System.Net.Sockets*) alla porta 54321. Successivamente avvia temporaneamente un *thread* secondario di ricezione che rimane in attesa dell'arrivo di dati, attraverso il metodo *UdpClient.Receive(),* ed eventualmente il memorizza in un *buffer*. Il *thread* principale interroga ciclicamente il *buffer* per un intervallo massimo di 5 secondi. Se viene ricevuto con successo almeno un datagramma entro il tempo prestabilito, la connessione viene considerata valida e il sistema procede all'inizializzazione degli indicatori di stato e delle strutture dati. In caso contrario, il sistema interrompe l'esecuzione del *thread* di ricezione e conclude la procedura senza stabilire alcuna connessione.

Nonostante il metodo proponga un approccio sequenziale per la ricerca di una delle modalità di connessione, è possibile, in fase di sviluppo e debugging, escludere quelle indesiderate tramite appositi *flag* (variabili booleane). Durante l'intero processo vengono aggiornati degli indicatori booleani, accessibili da metodi esterni, i quali consentono di monitorare lo stato della ricerca e il suo esito in tempo reale. Tali indicatori sono stati implementati con l'attributo *volatile*, garantendo che il valore della variabile sia sempre letto e scritto direttamente nella memoria principale, in modo tale da permettere la visibilità immediata del valore aggiornato tra *thread* diversi. Inoltre, è stato implementato un evento pubblico che viene generato ogni volta che lo stato della

connessione con la SyncStation o con le sonde Due+ subisce una variazione. L'argomento dell'evento rappresentante i 4 possibili stati della connessione: connesso, non connesso, in acquisizione o acquisizione terminata. Questo meccanismo consente alle altre componenti dell'applicazione di aggiornare in modo immediato il proprio comportamento in funzione dello stato corrente del sistema.

Avvio della trasmissione

Stabilita la connessione è necessario inviare dei comandi di controllo specifici per iniziare o terminare l'acquisizione dei segnali EMG. Tale gestione è implementata nel metodo *custom StartAcquisition*(), il quale si occupa di iniziare il flusso dati, in modo coerente con la modalità di comunicazione attiva, utilizzando i protocolli di comunicazione previsti dai produttori [111]. Nel caso della SyncStation, il comando consiste in una sequenza di byte costruita dinamicamente sulla base del numero di dispositivi abilitati. Il primo byte è uno *start byte*, in cui il bit meno significativo viene posto a 1 per indicare l'inizio dell'acquisizione, mentre i bit più significativi rappresentano la dimensione del comando. Seguono uno o più *control byte*, ciascuno relativo a una sonda attiva. Ogni control byte incorpora diversi campi informativi, tra cui l'ID della sonda (nei bit 7–4), il tipo di segnale da acquisire (bit 3, nel caso specifico EMG), e un bit di abilitazione del dispositivo (bit 0). L'ultimo byte del comando è un CRC8 (*Cyclic Redundancy Check* a 8 bit), esso rappresenta il resto della divisione logica (XOR) tra la sequenza di bit del messaggio e la stringa binaria 100110001 e viene utilizzato dalla SyncStation per verificare l'integrità del pacchetto ricevuto [111]. Una volta costruito, il comando viene trasmesso tramite lo *stream* TCP associato alla connessione stabilita precedentemente. Per interrompere l'acquisizione viene inviata una sequenza di due byte: uno *start byte* con valore 0x00 e un byte di CRC [111]. Esclusivamente in questa modalità è possibile inviare un comando di configurazione della latenza, utilizzato per impostare un valore numerico compreso tra 1 e 200. Il comando è composto da tre byte: uno *start byte* (bit 7 a 1, bit 5-1 rappresentanti il numero di *option bytes*), un *option byte* contenente il valore di latenza, e infine il CRC8 [111]. Nell'applicazione la latenza è stata impostata al valore minimo, al fine di ricevere i pacchetti Wi-Fi provenienti dalle sonde nel minor tempo possibile, ovvero ogni 27 ms nel caso ideale [110], evitando attese superflue dovute ai dati dei canali accessori non utilizzati nel progetto.

Nel caso della singola sonda, l'avvio dell'acquisizione avviene attraverso l'invio di un singolo byte di controllo. In questo contesto il valore inviato è 0x09. Tale valore rappresenta il comando start definito dal protocollo interno delle sonde Due+. In particolare, il bit 0 attiva l'acquisizione, mentre il bit 3 specifica la tipologia di segnale da acquisire, gli altri bit non utilizzati restano a zero. Poiché la comunicazione avviene direttamente con la sonda, non viene impiegato un CRC. Per terminare l'acquisizione dei dati viene inviato il byte 0x08 [111].

Nella modalità *stand-alone*, l'avvio e l'arresto dell'acquisizione sono gestiti dal PC, mentre l'applicazione sul visore si limita ad avviare o chiudere il *thread* di ricezione dall' UDP *socket*.

Lettura e decodifica dei dati

Dopo l'invio degli appositi comandi viene avviato un *thread* (*ReadDataLoop*() o *ReceiveData*()) dedicato alla lettura e decodifica dei dati. Il metodo implementa un ciclo continuo

(*loop while)* di lettura, che ad ogni iterazione verifica la presenza di dati nel *stream TCP* attraverso il metodo *DataAvailable* della classe *NetworkStream*. Se non ci sono dati disponibili per la lettura il *thread* viene messo in pausa per 1 ms. In caso contrario, il metodo interroga la proprietà *TcpClient.Available* per determinare quanti byte sono effettivamente presenti nel *buffer* di ingresso. Il numero di byte letti viene allineato al più grande multiplo della dimensione del campione, scartando gli eventuali byte residui che non consentirebbero la ricostruzione corretta di un campione completo. La dimensione del campione è determinata dal numero di canali trasmessi, in particolare ogni sonda attiva fornisce 8 canali a 16 bit a cui vanno sommati i 6 canali accessori della SyncStation. Il metodo registra inoltre il tempo di trasmissione dei pacchetti tramite uno *Stopwatch* (libreria *System.Diagnostics*) ad elevata precisione.

Una volta letti i dati, il metodo suddivide il pacchetto Wi-Fi in singoli campioni e, per ciascun dispositivo attivo, richiama la funzione *ParseSondaData*(). Questo metodo scompone (*parsing*) e decodifica la sequenza di 16 byte degli 8 canali trasmessi dalla sonda in variabili numeriche. I dati elettromiografici presenti nei primi due canali (primi 4 byte), vengono estratti convertendo le coppie di byte in interi a 16 bit con segno tramite il metodo *BitConverter.ToInt16* (libreria *System*). I valori ottenuti sono moltiplicati per un fattore di conversione che consente di trasformare il campione digitale nella tensione reale rilevata. Tale fattore è stimato dividendo la risoluzione del convertitore analogico-digitale (ADC) per il guadagno applicato prima della conversione [110]. La risoluzione dell'ADC è definita come il rapporto tra la dinamica di ingresso (3.3 V) e il numero di livelli disponibili (2^16), considerando un guadagno di 202, si ottiene un valore pari a 0,000249 mV [110]. I quattro canali successivi rappresentano i quaternioni derivati dal sensore inerziale. Dopo la conversione in interi a 16 bit, vengono normalizzati nell'intervallo [-1, 1] dividendoli per il fattore 8192, ovvero 2^13, in quanto la risoluzione reale del sensore è 14 bit con segno (cfr. sez. "Caratteristiche dei sensori"). La normalizzazione è necessaria per la successiva conversione in angoli di Eulero (*Roll, Pitch e Yaw*) implementata utilizzando il metodo *eulerAngles* della classe *Quaternion* di Unity. Infine, la funzione estrae i due canali accessori: il livello di riempimento del *buffer* interno della sonda e un contatore di campioni incrementale utilizzato per verificare eventuali perdite di dati durante la trasmissione. Al termine della decodifica, tutti i valori estratti vengono inseriti in un'apposita struttura dati.

Il *thread* dedicato alla modalità di connessione *stand-alone* adotta una logica analoga a quella utilizzata per la lettura via TCP. Avvia un ciclo continuo di ascolto che attende la ricezione dei datagrammi UDP provenienti dal PC, dai quali estrae tutti i campioni contenuti per inserirli nei rispettivi *buffer* di acquisizione. In questa modalità i dati ricevuti, che rappresentano l'inviluppo del segnale EMG, sono codificati in formato binario a 32 bit e vengono convertiti direttamente in valori *floating-point* (*Single*) tramite il metodo *built-in BitConverter.ToSingle*. Anche per questa modalità viene stimata la frequenza di arrivo dei datagrammi, così da monitorare la stabilità e la regolarità della trasmissione.

In entrambi i casi i *thread* vengono chiusi quando viene chiamato il metodo per terminare l'acquisizione (*StopAcquisition*()), impostando a *false* la condizione del *loop*. Qualora la variazione del *flag* non sia sufficiente a interrompere il *thread* (situazione che si verifica potenzialmente se il processo rimane bloccato in un'operazione di lettura sul *socket*) il sistema

adotta una strategia di terminazione forzata. Attraverso il metodo *Join()* della classe *Thread* (libreria *System.Threading*), il software attende la chiusura spontanea per un intervallo massimo di 1 s, se questo non avviene, vengono invocati esplicitamente i metodi *Close()* delle classi *TcpClient* o *UdpClient* (a seconda della modalità operativa) e *NetworkStream* mediante la funzione *custom ResetConnectionState()*. Questa azione interrompe forzatamente il canale di comunicazione a livello hardware, scatenando un'eccezione controllata che obbliga il *thread* a uscire dal loop di ricezione. Per avviare nuovamente l'acquisizione dopo l'interruzione forzata è necessario stabilire nuovamente la connessione tramite l'apposito metodo.

L'organizzazione e l'elaborazione dei dati provenienti dalle sonde Due+ sono affidate alla classe *custom DuePlusDataContainer*. Tale classe mantiene internamente una serie di liste pubbliche che tengono traccia di tutti i campioni acquisiti, alle quali i metodi dell'applicazione possono accedere. La classe *SyncStationConnection* crea una lista di oggetti *DuePlusDataContainer* pari al numero di dispositivi Due+ attivi. In questo modo, ciascuna sonda dispone di un proprio contenitore dedicato, garantendo una corrispondenza tra i dati ricevuti e il dispositivo da cui provengono. In caso di connessione da remoto, il *DuePlusDataContainer* è istanziato con la variabile *apply_preprocessing* impostata a *false*. In questo modo i dati passati alla classe non vengo processati, ma salvati direttamente nelle apposite liste dell'inviluppo. L'inserimento dei dati avviene attraverso il metodo custom *Add*(), che riceve in ingresso i valori estratti dal pacchetto di acquisizione e li appende alle rispettive liste. In questo modo ogni *DuePlusDataContainer* rappresenta un *buffer* completo per una singola sonda, contenente l'intera sequenza temporale dei segnali registrati durante l'esecuzione dell'esercizio.

Pre-processing e salvataggio dei dati

Il pre-processing dei dati è implementato per ottenere l'inviluppo del segnale EMG sottocampionato a 50 HZ per materne la coerenza con l'aggiornamento del motore fisico di Unity. Gli step del filtraggio sono:

- Filtraggio Notch: per eliminare l'interferenza della frequenza di rete elettrica a 50 Hz.
- Rettifica del segnale EMG: dal segnale filtrato viene estratto il valore assoluto (|x|, dove x è il segnale EMG filtrato), trasformando le deflessioni negative in positive.
- Estrazione dell'Inviluppo: il segnale rettificato viene processato da un filtro Butterworth del 2° ordine con una frequenza di taglio a 8 Hz.
- Sottocampionamento: l'inviluppo del segnale viene sottocampionato da 2000 Hz a 50 Hz selezionando un campione ogni 40.

Nella modalità PC-Link le operazioni di filtraggio sono state implementate utilizzando la libreria *MathNet.Filterig*. La classe *DuePlusDataContainer* gestisce il flusso dei dati inizializzando, nel metodo costruttore, due istanze della classe *EMGEnvelope* per il calcolo dell'inviluppo e i filtri notch ricorsivi.

La funzione *custom BuildRecursiveFilterCoefficients* calcola i coefficienti del filtro notch a partire dalla frequenza di campionamento (f_c = 2000Hz) e dalla frequenza di taglio (f_t = 50Hz). Definendo il periodo di campionamento $T = 1/\mathrm{fc}$ i coefficienti del numeratore del filtro sono [112]:

$$b_0 = 1; \qquad b_1 = -2(1 - c_2)\cos(2\pi f_t T); \quad b_2 = (1 - c_2)^2 \tag{1}$$

Mentre quelli del denominatore:

$$a_0 = 1; \qquad a_1 = -2(1 - c_1)\cos(2\pi f_t T); \quad a_2 = (1 - c_2)^2 \tag{2}$$

Con i parametri

$$c_1 = \pi * B * T; \qquad c_2 = c_1 * z \tag{3}$$

Dove B (impostato a 2) e z (impostato a 0.01) controllano rispettivamente la larghezza di banda e l'attenuazione del filtro.

Nel metodo *Add*() i campioni EMG vengono accumulati in un *buffer* temporaneo. Quando la dimensione del *buffer* raggiunge i 40 campioni, viene invocato il metodo *CalculateEnvelope*, il quale applica i filtri notch e trasmette i dati alle due istanze della classe *EMGEnvelope* per il calcolo dell'inviluppo. Quest'ultimo è ottenuto mediante il metodo *Envelope*(), nel quale i nuovi campioni EMG vengono utilizzati per riempire un *buffer* FIFO (First-In-First-Out) di dimensione prestabilita (272 campioni), successivamente, il segnale viene rettificato e filtrato passa-basso a 8Hz. Al termine del calcolo, il metodo restituisce l'ultimo campione dell'inviluppo e rimuove i primi 40 campioni dal *buffer*, effettuando così un sottocampionamento a 50 Hz e riducendo gli effetti transitori grazie alla presenza della finestra mobile.

Il salvataggio dei dati è gestito dalla classe *SyncStationConnection*, che attraverso la funzione *SaveData*() salva la frequenza di ricezione dei dati (metodo *SaveTime*()) e richiama per ogni contenitore creato il metodo *custom SaveDataToFile*() passandogli in ingresso il percorso di salvataggio e il nome del file. I dati vengono archiviati in formato CSV all'interno di due file distinti per ogni sonda attiva, identificati univocamente tramite il rispettivo canale della SyncStation (es. CH7, CH8). Questo approccio garantisce che la logica di salvataggio rimanga costante indipendentemente dal numero di dispositivi collegati, evitando l'incremento del numero di colonne a favore di una generazione di set di file indipendenti per ogni sensore. Nello specifico, il file raw.csv contiene i segnali bioelettrici grezzi e i parametri inerziali in una struttura a sette colonne (EMG Ch1, EMG Ch2, *Roll, Pitch, Yaw, SampleCounter e BufferUsage*), mentre il file env.csv è dedicato agli inviluppi del segnale EMG dei due canali, organizzati in due colonne (Ch1 e Ch2). Parallelamente, viene generato un unico file denominato Sampling_Time.csv contenente una sola colonna relativa ai tempi di campionamento.

Procedura di calibrazione

Nello scenario virtuale proposto, il gesto della remata è mappato sulla traiettoria lineare del movimento di *reaching* eseguito dal paziente, la cui implementazione tecnica è discussa nel paragrafo "Task motorio e logica di controllo dei remi". Il *Range Of Motion* (ROM) di tale movimento deve essere calibrato sulla massima estensione eseguibile dal paziente, per due motivi fondamentali: far esercitare il paziente a pieno titolo, sfruttando tutte le sue capacità residue, per stimolare i processi di plasticità neurale e *motor learning* [32]; garantire una partecipazione paritaria all'interno del compito collaborativo, assicurando che ogni utente possa fornire un contributo alla propulsione della barca equivalente a quello dei compagni, indipendentemente dall'entità del deficit motorio. Infatti, il completamento di una "voga virtuale" non è vincolato al raggiungimento di una distanza fisica standard, ma è calibrato sul ROM residuo del singolo soggetto.

A tal fine, l'architettura software integra una specifica Scena Unity[2] volta a definire i parametri spaziali di inizio e fine del movimento riabilitativo. In questa fase, qualora il protocollo includa il monitoraggio elettromiografico, viene stabilita automaticamente la connessione con i dispositivi di acquisizione e registrato il valore di Massima Contrazione Volontaria (MVC) del deltoide laterale. Tale parametro è finalizzato alla definizione delle soglie operative impiegate dal sistema di biofeedback per l'individuazione di attivazioni muscolari improprie, i cui dettagli verranno approfonditi nei capitoli successivi. Inoltre, durante la procedura è possibile selezionare la lingua dell'applicazione scegliendo tra italiano o inglese.

Un aspetto fondamentale è che all'inizializzazione dell'applicazione Unity, il sistema definisce l'origine dello spazio di tracciamento in base alla posizione istantanea del visore. Poiché le coordinate spaziali dei controller sono calcolate in modo relativo a tale origine, la coerenza dei dati cinematici è strettamente dipendente dalla stabilità posturale dell'utente. È pertanto fondamentale che il soggetto mantenga la medesima posizione seduta per l'intera durata della sessione; un eventuale spostamento del tronco comporterebbe infatti l'introduzione di un offset sistematico nelle misure di *reaching* inficiando la precisione dei biofeedback.

[2]In Unity, la Scena è un contenitore che definisce un ambiente isolato all'interno del progetto. Ogni Scena contiene la propria gerarchia di elementi rappresentando un modulo distinto dell'applicazione. Unity permette di passare in maniera dinamica da una Scena all'altra (es. fase di calibrazione → menù → lobby → ambiente di gioco).

Panoramica della procedura

La calibrazione è la prima operazione che effettua l'utente quando viene avviata l'applicazione, ed è svolta in locale, ovvero prima della connessione con il Game Server, che avviene quando la procedura è terminata cliccando un apposito pulsante.

All'avvio il sistema consente all'utente di selezionare la lingua desiderata, successivamente tenta di stabilire la connessione con i dispositivi di acquisizione EMG, in una delle due modalità previste (cfr. sez. "Modalità operative previste dal sistema di teleriabilitazione"). Nel caso in cui non venga rilevato nessun dispositivo connesso, la procedura viene avviata escludendo la parte relativa all'acquisizione dell'MVC e il gioco non prevederà l'utilizzo del segnale EMG come biofeedback. Proseguendo l'utente visualizza la schermata principale in cui sono presenti pulsanti che consentono di accedere alle diverse fasi della calibrazione e un feedback testuale che guida l'utente durante l'intero processo. Il sistema abilita progressivamente i pulsanti dell'interfaccia, consentendo all'utente di procedere solo secondo il flusso previsto e impedendo sequenze non coerenti.

Le fasi previste durante la calibrazione sono:

- **Valutazione del ROM residuo**: viene effettuata richiedendo al paziente il mantenimento statico del controller in tre configurazioni spaziali specifiche, ciascuna per un intervallo di sei secondi. La prima acquisizione avviene posizionando il dispositivo in corrispondenza del petto, fungendo da riferimento essenziale per compensare eventuali offset cinematici introdotti dalla posizione relativa dei controller rispetto al visore. Successivamente, vengono registrate le coordinate spaziali soggetto-specifiche relative al punto iniziale e al punto finale del movimento riabilitativo. Le istruzioni relative alle tre posizioni vengono mostrate preventivamente in un *pop-up* informativo, e successivamente riproposte durante la calibrazione, in modo da fornire un supporto visivo e testuale che accompagna l'utente lungo tutta la procedura
- **Stima della MVC:** qualora nella fase precedente sia stato rilevato un dispositivo per l'acquisizione del segnale EMG, l'utente dovrà eseguire una prima fase di acquisizione del rumore di fondo e successivamente tre contrazioni massimali del deltoide laterale. La fase iniziale prevede il monitoraggio dell'arto in una condizione di completo rilassamento per una durata di 5 secondi, operazione necessaria per definire l'offset elettrico e le interferenze ambientali. La procedura per la stima dell'MVC consiste nell'esecuzione di tre prove di abduzione isometrica sul piano frontale, mantenendo l'arto con un angolo di circa 90 gradi rispetto al tronco per una durata di 8 secondi per ciascuna ripetizione. Tra una ripetizione e l'altra è prevista una fase di riposo di un 45 secondi, per evitare l'insorgenza di fenomeni di fatica. La contrazione massimale può essere svolta utilizzando una superficie rigida, ad esempio un tavolo, per bloccare l'elevazione del braccio, alternativamente è possibile adoperare delle bande rigide o elastiche (ad elevata resistenza) opportunamente ancorate oppure avvalendosi della resistenza manuale da parte di un caregiver. L'intero processo è supervisionato da indicatori visivi che regolano i tempi di contrazione e riposo.

- **Verifica della calibrazione:** terminata la calibrazione, l'utente ha la possibilità di verificarne la correttezza eseguendo il movimento riabilitativo. In questa fase, la mano virtuale visualizzata nell'interfaccia deve riprodurre fedelmente i movimenti reali, spostandosi coerentemente tra il punto iniziale e quello finale. Nel caso in cui la sessione preveda l'impiego del monitoraggio EMG, l'utente è invitato a effettuare delle contrazioni volontarie del deltoide (movimenti di abduzione e adduzione) per verificare che il muscolo virtuale si attivi in maniera coerente con l'attività muscolare reale. Questa fase permette di confermare la coerenza tra la calibrazione eseguita e la risposta effettiva del sistema, fornendo un riscontro immediato sulla correttezza della configurazione prima dell'avvio dell'esercizio riabilitativo. Se la calibrazione risulta errata, è possibile ripeterla mediante l'apposito pulsante, oppure confermarla, con conseguente salvataggio dei dati.

Descrizione dei Game Object principali

I Game Object (GO) presenti nella Scena di calibrazione sono mostrati in Figura 17.

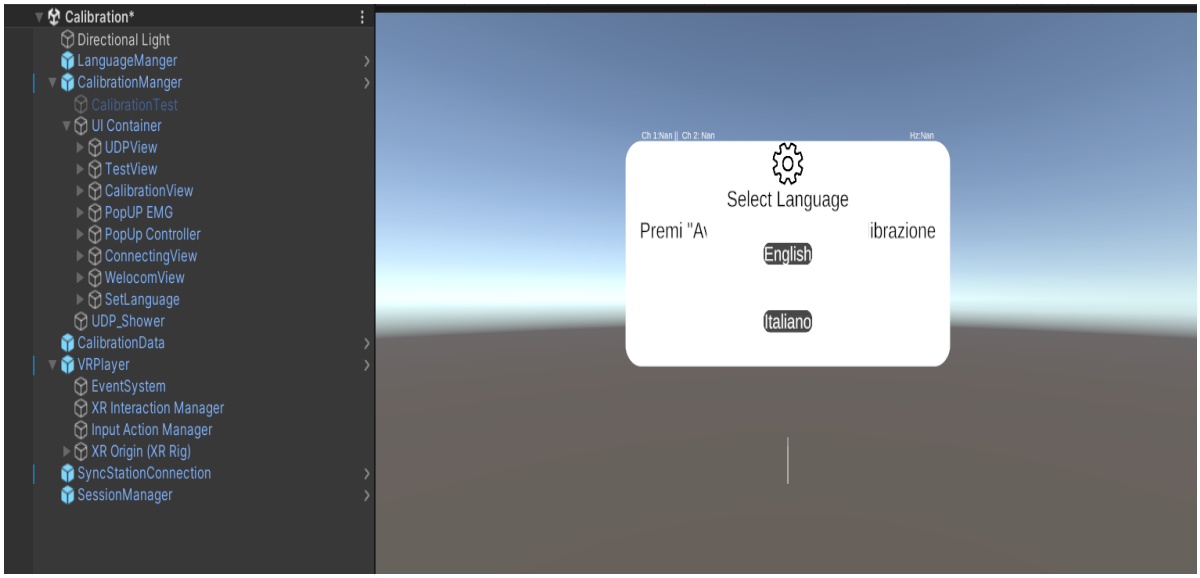

Figura 17: Game Object presenti nella Scena di calibrazione

Di seguito sono descritte le funzioni dei vari GO:

- ***LanguageManager*** è il componente dedicato alla gestione multilingua dell'interfaccia utente, contiene tutti i testi della UI (etichette, pulsanti, messaggi e istruzioni operative) in due lingue, italiano e inglese, organizzati sotto forma di dizionari. I vari componenti della Scena possono accedere al dizionario corrente utilizzando una chiave, che sarà la stessa per entrambi i dizionari. In questo modo è possibile ottenere una scritta in maniera dinamica a seconda del dizionario in uso in quell'istante.
- ***CalibrationData*** ha il compito di raccogliere, elaborare e archiviare i dati acquisiti durante le diverse fasi della calibrazione. In particolare, memorizza le coordinate dei

controller e i segnali EMG provenienti dai sensori, elaborandoli per ottenere parametri sintetici utili alle fasi successive del gioco.

- ***SyncStationCommunication*** è responsabile della connessione e comunicazione con i dispositivi EMG. Si occupa inoltre di decodificare, archiviare e salvare i dati provenienti dai sensori, rendendoli disponibili agli altri componenti dell'applicazione.
- ***SessionInfo*** contiene le informazioni utili per le fasi successive dell'applicazione, come il nome univoco della sessione in corso per il salvataggio dei dati, composto da una stringa, (nome della sessione o dell'utente) unita alla data e ora della sessione in corso e una variabile booleana che indica se è stata rilevata una connessione con i dispositivi EMG.
- ***CalibrationManager*** costituisce l'elemento logico principale nella procedura di calibrazione. La sua funzione è quella di coordinare l'intero flusso operativo, gestendo la connessione e l'acquisizione dei dati dai dispositivi EMG, aggiornando dinamicamente le interfacce grafiche e i feedback testuali e coordinando il flusso di dati tra i vari GO. Contiene come oggetto "figlio" il *CalibrationTest*, il quale implementa la logica per il movimento delle mani virtuali, elabora il segnale elettromiografico per determinare le attivazioni muscolari e aggiorna la UI dedicata in tempo reale, fornendo un feedback visivo durante la fase di verifica della calibrazione.
- ***UDP_Shower*** mostra nella UI i dati provenienti dai dispositivi EMG e la frequenza di trasmissione di quest'ultimi.
- **UIContainer** costituisce l'elemento di riferimento per la posizione delle UI nella Scena, garantendo un posizionamento sempre frontale rispetto alla visuale del giocatore ad una distanza ottimale per la visualizzazione delle interfacce. Organizzando le diverse interfacce come Game Object "figli" all'interno di questo contenitore, il sistema sfrutta la funzionalità gerarchia di Unity, nella quale ogni elemento annidato eredita automaticamente la posizione e l'orientamento globale del componente padre. Di conseguenza, è possibile traslare e orientare l'intero set di interfacce agendo su un unico punto di controllo anziché sulle singole componenti.
- ***VRPlayer***, descritto nel capitolo precedente (cfr. sez. "Integrazione Meta Quest 3), gestisce la presenza e l'interazione dell'utente nello spazio virtuale.

All'interno dello scenario virtuale la maggior parte delle classi *custom* associate ai GO appena descritti, non eseguono istruzioni in modo continuo tramite il metodo *Update*, fatta eccezione per il *CalibrationTest* e *l'UDPShower*. La maggior parte dei componenti reagisce invece agli input dell'utente o agli eventi generati dal sistema. Quando l'utente interagisce con un pulsante, è il *CalibrationManager* a richiamare i metodi appropriati degli altri GO e ad aggiornare dinamicamente l'interfaccia. Per gestire le diverse fasi della calibrazione, che richiedono intervalli temporali precisi e sequenze di attesa, il sistema si avvale delle *coroutine*. In Unity, una *coroutine* è un metodo speciale, di tipo *IEnumerator*, capace di sospendere la propria esecuzione e restituire il controllo al flusso principale, per poi riprendere esattamente dal punto in cui si era interrotta nel *frame* successivo. Attraverso l'istruzione *yield return*, la *coroutine* può essere messa in pausa per una durata specifica o fino al verificarsi di una condizione.

Questa caratteristica è fondamentale per non bloccare l'esecuzione dell'applicazione durante i cicli di attesa, a differenza di una funzione standard che deve essere completata interamente all'interno di un singolo *frame*.

Alcuni componenti fondamentali dell'applicazione, quali *LanguageManager*, *SyncStationConnection*, *CalibrationData* e *SessionInfo*, sono stati implementati come *Singleton*. Un *Singleton* in Unity garantisce che una classe abbia una sola istanza e fornisce un punto di accesso globale ad essa. Dal punto di vista implementativo, vengono realizzati tramite la definizione di un riferimento pubblico e statico denominato *Instance*. Durante la fase di inizializzazione nel metodo nativo *Awake*(), il componente esegue un controllo di integrità: qualora venga rilevata una preesistente istanza della medesima classe (eventualità comune durante le transizioni tra diverse Scene), l'oggetto ridondante viene rimosso per preservare l'integrità del riferimento globale e prevenire conflitti nei dati. Questa scelta progettuale ottimizza l'interazione tra i moduli del sistema: ogni classe può infatti interrogare le variabili o invocare i metodi dei *Singleton* tramite la sintassi NomeClasse.Instance. Ciò elimina la necessità di stabilire riferimenti espliciti *nell'Inspector* o di ricorrere a operazioni di ricerca dei componenti (come *GameObject.Find*).

Per estendere la persistenza di questi componenti durante l'intero ciclo di vita dell'applicazione, viene impiegato il metodo *built-in DontDestroyOnLoad*(), il quale impedisce la distruzione del Game Object associato durante il caricamento di nuovi scenari.

Implementazione delle interfacce utente in Unity

La realizzazione delle interfacce grafiche si basa sul sistema Unity UI (uGUI). Il nucleo di ogni interfaccia è rappresentato dal componente Canvas, il quale funge da superficie 2D sulla quale vengono renderizzati tutti gli elementi UI. Il Canvas deve essere associato ad un GO della Scena e tutti gli elementi dell'interfaccia utente devono essere "figli" di tale oggetto. Unity fornisce una serie di componenti UI predefiniti sia interattivi che visivi con un elevato grado di personalizzazione. Gli elementi interattivi come bottoni, caselle selezionabili, menù a tendina, cursori e campi di immissione hanno delle funzionalità integrate per visualizzare le transizioni tra gli stati (normale, evidenziato, premuto, disabilitato) e per la navigazione verso altri elementi selezionabili utilizzando la tastiera o il controller. I componenti di interazione hanno almeno un *UnityEvent* che viene richiamato quando l'utente interagisce con il componente, eseguendo i metodi associati a quella specifica interazione, sia in fase di sviluppo tramite *l'Inspector*, sia dinamicamente durante l'esecuzione dell'applicazione. I componenti visivi sono testi o immagini che possono essere visualizzate nell'interfaccia, in particolare ogni immagine ha una *Rect Transform* che ne determina le dimensioni e la posizione nell'interfaccia, mentre l'immagine sorgente deve essere convertita in *UI sprites*.

Nello scenario virtuale, il sistema UI non è vincolato allo schermo (modalità *Screen Space*), ma è integrato nello spazio tridimensionale (modalità *World Space*). L'interazione è resa possibile dal *Tracked Device Graphic Raycaster*, un modulo *dell'XR Interaction Toolkit* che permette alle interfacce di rispondere ai raggi laser emessi dai controller. *L'XR Interaction Toolkit* è impiegato anche per garantire il corretto posizionamento delle interfacce attraverso la classe *LazyFollow.*

Questo algoritmo assicura che il contenitore delle interfacce (*UI_Container*) segua dolcemente i movimenti del visore evitando spostamenti bruschi che potrebbero compromettere l'ergonomia visiva e indurre disagi all'utente. Il componente consente di definire delle soglie personalizzabili, attraverso le quali lo spostamento dell'interfaccia viene innescato o interrotto in base alla rotazione o traslazione del capo, garantendo una stabilità visiva durante le naturali micro-oscillazioni della testa e premettendo al soggetto di esplorare l'ambiente circostante.

Per una gestione efficiente della visibilità e dell'interazione, ogni modulo UI è stato equipaggiato con l'elemento *built-in Canvas Group* e uno script *custom UI_View*. Il *Canvas Group* permette di manipolare automaticamente lo stato di un intero gruppo di elementi, controllandone l'opacità (parametro *alpha*), la reattività agli input (booleano *interactable*) e la capacità di intercettare i puntatori dei controller VR (booleano *blocks raycasts*). Tale meccanismo è gestito dallo script *UI_View*, il quale implementa i metodi *Show*() e *Hide*(). Queste funzioni modificano opportunamente i parametri sopra citati, impostando rispettivamente il valore di *alpha* a 1 e i booleani a *true* per rendere l'interfaccia visibile e funzionale, oppure *alpha* a 0 e i booleani a *false* per nasconderla e disabilitarla. La classe *UI_View* viene utilizzata per referenziare le diverse interfacce all'interno del *CalibrationManager*, il quale ne coordina l'attivazione dinamica in base alle necessità del flusso operativo.

Parallelamente, il sistema amministra il livello di interattività dei singoli elementi grafici per garantire un'esecuzione rigorosa del protocollo, agendo direttamente sulla proprietà *interactable* dei pulsanti, attraverso i metodi *SetButtonsInteractable*() e *UpdateButtonStates*() del *CalibrationManager*. Nello specifico, all'inizio di ogni procedura viene invocato *SetButtonsInteractable*(), con il parametro di ingresso impostato a *false*, una misura precauzionale che disattiva tutti i pulsanti al fine di impedire input accidentali che genererebbero conflitti durante l'esecuzione dell'applicazione. Solo al termine, la logica di *UpdateButtonStates*() provvede a una riattivazione selettiva e condizionale dei comandi, interrogando i *flag* di avanzamento del sistema quali *isControllerCalibrated*, *isEMGCalibrated* e *isTestDone*. Al contempo, il sistema provvede all'aggiornamento dinamico dei feedback testuali per guidare l'utente lungo tutta la sequenza operativa predefinita. Questo meccanismo di validazione assicura che l'accesso alle fasi successive, come il test della calibrazione o l'avvio dell'esercizio, sia subordinato al completamento delle operazioni precedenti, riducendo drasticamente il rischio di stati incoerenti tra l'interfaccia e la logica applicativa, garantendo che tutti i parametri critici siano correttamente inizializzati prima del passaggio allo scenario di gioco.

Interfacce utente e logica di interazione

Scelta della lingua e aggiornamento dinamico delle scritte

La prima interfaccia che visualizza l'utente è mostrata in Figura 18, ed è composta da due pulsanti, un testo e un'icona delle impostazioni. Da questa interfaccia è possibile selezionare una delle due lingue, cliccando su uno dei due pulsanti.

Figura 18: Interfaccia SetLanguage e relativi GO.

Dal punto di vista implementativo la gestione della lingua è coordinata dai componenti *custom LanguageManager*, *CalibrationManager, LocalizedTextTMP e LKeys*.

LKeys è una classe statica di supporto, che contiene le chiavi di accesso alle stringhe dei dizionari linguistici, alla quale tutti i componenti possono accedere. Inoltre, raggruppa le enumerazioni pubbliche, tra cui *AppLanguage* e *UIKey*, il primo specifica le lingue supportate (*AppLanguage.Italian* o *AppLanguage.English)*, mentre il secondo contiene i riferimenti per i testi statici utilizzati nell'interfaccia utente. In C#, l'enumerazione (*enum*) è un tipo di dato che definisce un insieme di costanti nominali, il suo impiego previene errori logici garantendo che i parametri passati alle funzioni appartengano esclusivamente al set di valori predefinito.

L'interazione dell'utente con i pulsanti di selezione della lingua (Figura 18) attiva i metodi del *CalibrationManager: OnClickItaliano() e OnClickEnglish()*. Tali funzioni invocano a loro volta il metodo *SetLanguage*() del *LanguageManager*, passando come argomento rispettivamente il valore *Italian* e *English* dell'enumerazione *AppLanguage*.

SetLanguage() aggiorna la lingua corrente modificando il dizionario di stringhe attivo, il quale contiene le traduzioni di etichette, messaggi e testi informativi. Il *LanguageManager* implementa un evento pubblico (*OnLanguageChanged()*), generato ogni volta che la lingua viene modificata. L' evento funge da notifica globale, permettendo a tutti gli altri script sottoscritti di aggiornare automaticamente i propri contenuti testuali.

Per far sì che tutte le scritte vengano aggiornate simulatamente ogni elemento testuale delle interfacce (etichette dei pulsanti, istruzioni sulla calibrazione, ecc.) contiene il competente *LocalizedTextTMP*.cs, che gestisce l'aggiornamento automatico del testo.

Lo script contiene una chiave (*UIKey*) che identifica univocamente la stringa nel dizionario del *LanguageManager*. La classe si registra all'evento *OnLanguageChanged*, che una volta invocato, esegue il metodo *Refresh()*. Tale metodo aggiorna il contenuto testuale dell'interfaccia accedendo al dizionario linguistico corrente richiamando il metodo *GetText ()* dell'istanza del *LanguageManager* tramite la chiave assegnata durante la fase di sviluppo. Mediante questo approccio, ogni volta che l'utente cambia lingua, tutti gli elementi testuali della interfaccia vengono aggiornati simultaneamente.

Per la gestione dei testi dinamici, ovvero quelli che variano in base allo stato della calibrazione o del sistema, è il *CalibrationManager* che all'occorrenza accede direttamente alla scritta nel dizionario richiamando sempre il metodo *GetText ()* con la chiave opportuna.

Connessione con i dispositivi EMG

Selezionato uno dei due pulsati, ed eseguiti i relativi metodi, il *CalibrationManager* nasconde l'UI e abilita l'interfaccia di benvenuto (WelcomeView*,* Figura 19).

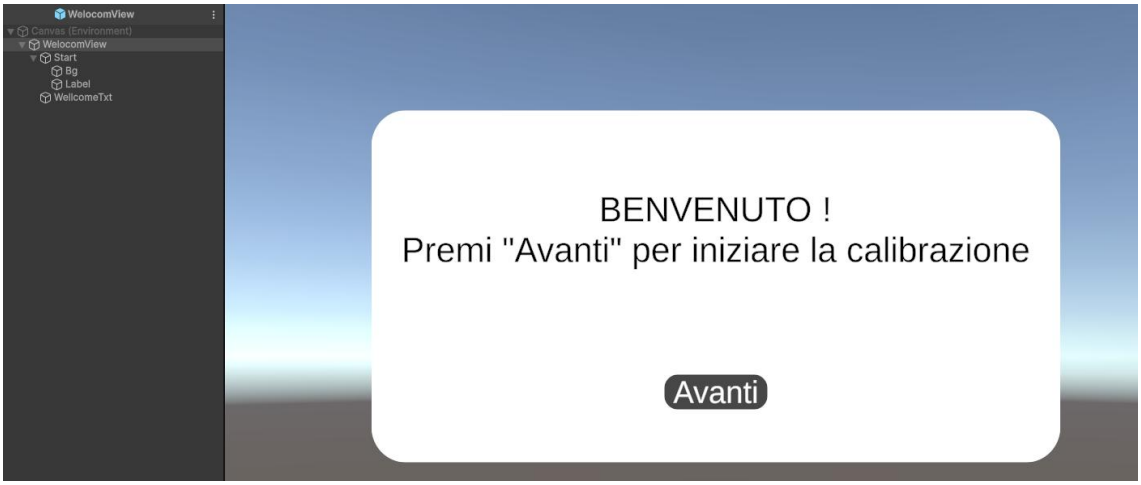

Figura 19: Interfaccia WelcomeView e relativi GO.

Cliccando il pulsante "AVANTI", il *CalibrationManager* esegue il metodo *OpenConnection()*, il quale nasconde la *WelcomeView*, attiva la schermata successiva di caricamento *(ConnectingView,* Figura 20) disabilitando i relativi pulsanti e richiama la *coroutine ConnectSyncStationWithLoading()*. Tale *coroutine* avvia l'animazione grafica dello *spinner* di caricamento, richiama il metodo *StartConnection()* della classe *SyncStationCommunication (cfr. sez. "*Connessione e comunicazione con le sonde*"*) e attende il completamento della connessione. Parallelamente, informa l'utente in tempo reale sullo stato della ricerca, interrogando gli appositi indicatori di *SyncStationCommunication* e aggiornando coerentemente il campo testuale dell'interfaccia con le scritte "Tentativo di connessione alla SyncStation...", "Tentativo di connessione con la probe singola..." e "Tentativo di connessione remota con la probe singola...". Appena viene rilevata la connessione con un dispositivo o al termine della ricerca, l'utente viene informato se è stata stabilita una connessione e in che modalità, viene fermata l'animazione dello *spinner* e vengono riattivati i pulsanti.

Figura 20: Interfaccia ConnectingView e relativi GO.

A questo punto cliccando il pulsante "AVANTI" della *ConnectingView* sarà possibile proseguire con la calibrazione oppure richiamare nuovamente il metodo *ConnectSyncStationWithLoading()* con il pulsante "RIPROVA CONNESSIONE". Nell'eventualità in cui l'utente decida di proseguire senza aver stabilito una connessione, tramite il metodo *OpenCalibrationView*() vengono aggiornati i *flag* interni del *CalibrationManager* per disabilitare le procedure che interessano i segnali EMG e viene mostrata la *CalibrationView* (Figura 21), ovvero l'interfaccia principale della calibrazione.

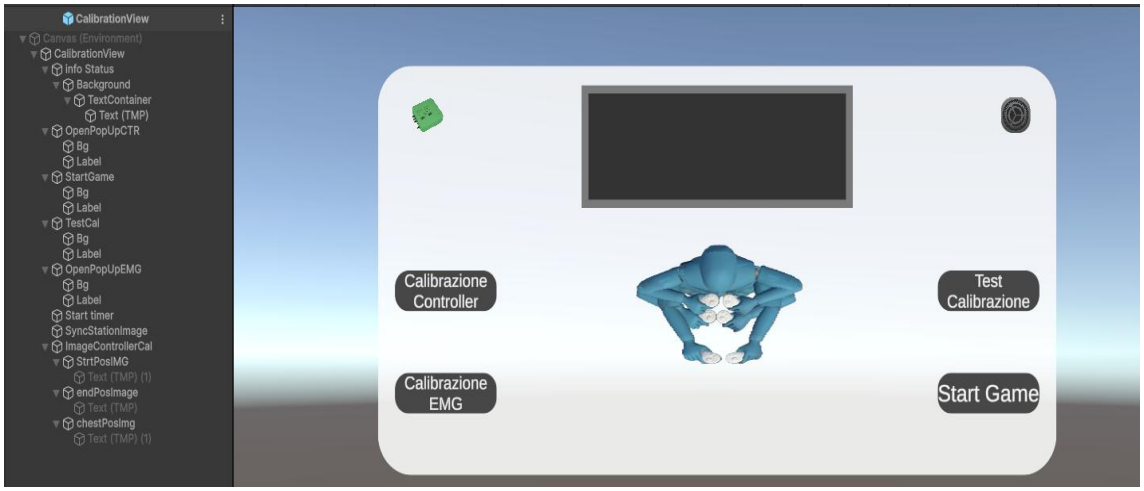

Figura 21: Interfaccia CalibrationView e relativi GO.

Questa interfaccia è composta da:

- 4 pulsanti che permettono di accedere alle 4 fasi della calibrazione.
- Un'immagine in alto a sinistra che rappresenta lo stato della connessione con i dispositivi EMG attraverso un codice a colori (rosso: non connesso, verde: connesso, giallo: in acquisizione, arancione: acquisizione in pausa, grigio: *default*).
- Un'icona delle impostazioni che permette di visualizzare nuovamente l'interfaccia SetLanguageView per cambiare la lingua.
- Un box testuale nel quale saranno visualizzate le istruzioni per l'utente.
- 3 immagini nella zona centrale che verranno attivate in sequenza per fornire un feedback visivo durante la calibrazione del ROM
- Un *timer*, sempre nella zona centrale, che fornisce all'utente le tempistiche durante la procedura.

Calibrazione ROM

Quando la CalibrationView viene mostrata, il *CalibrationManager* attiva solamente il pulsante "CALIBRAZIONE CONTROLLER" mentre gli altri risultano non interagibili. In aggiunta, nel box testuale è indicata l'istruzione che deve eseguire l'utente (Figura 22 A). Una volta che il pulsante viene premuto si apre l'interfaccia *PopUPController* (Figura 22 B), nella quale sono indicate le istruzioni che dovrà svolgere l'utente per effettuare la calibrazione e le immagini relative alle 3 posizioni da adottare. L'utente può tornare all'interfaccia precedente mediante il pulsante "ANNULLA", oppure avviare la calibrazione ciccando il pulsante "AVANTI", al quale è associato il metodo *StartCalibrationController() del CalibrationManager.*

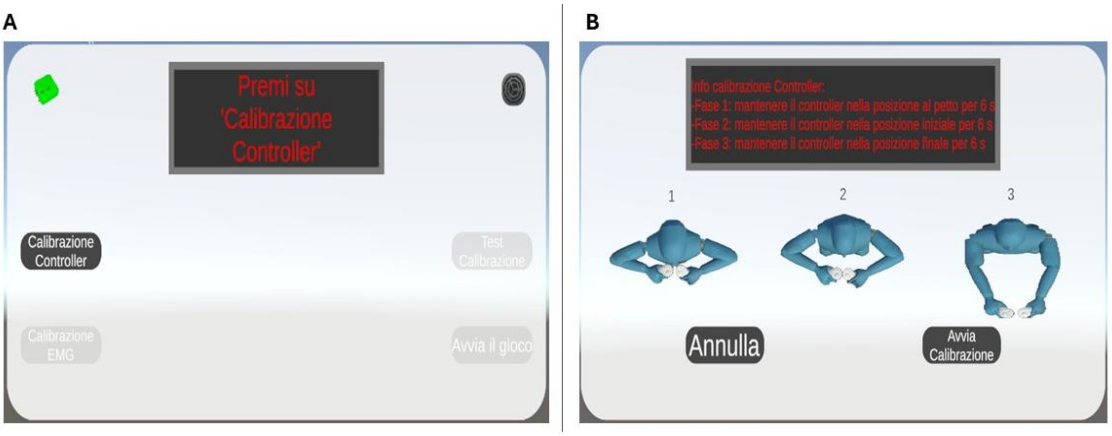

Figura 22: Interfacce utente per la calibrazione del ROM.A) Interfaccia CalibrationView nella fase iniziale della calibrazione. B) Pop-up informativo.

Tale funzione mostra nuovamente la *CalibrationView*, disattivando temporaneamente i pulsanti di interazione e chiama la *coroutine StartWithDelay()*. Quest'ultima avvia e mostra nell'interfaccia un conto alla rovescia, della durata del parametro in ingresso, 3 s in questo caso, per informare l'utente dell'imminente inizio della procedura (Figura 23 A).

Terminato il *countdown*, la gestione passa alla coroutine *CalibrateZ()*, responsabile dell'acquisizione delle posizioni lungo l'asse *z* dei controller destro e sinistro. All'interno di *CalibrateZ()*, l'utente è guidato nella corretta esecuzione delle tre pose previste dal protocollo grazie a un sistema integrato di feedback testuali e grafici. La *coroutine* regola dinamicamente la visualizzazione delle immagini guida, attivandole o nascondendole in funzione della fase corrente della calibrazione, e aggiorna i messaggi illustrativi per indicare quando mantenere la posizione o quando passare alla successiva (Figura 23 B, Figura 23 C, Figura 23 D).

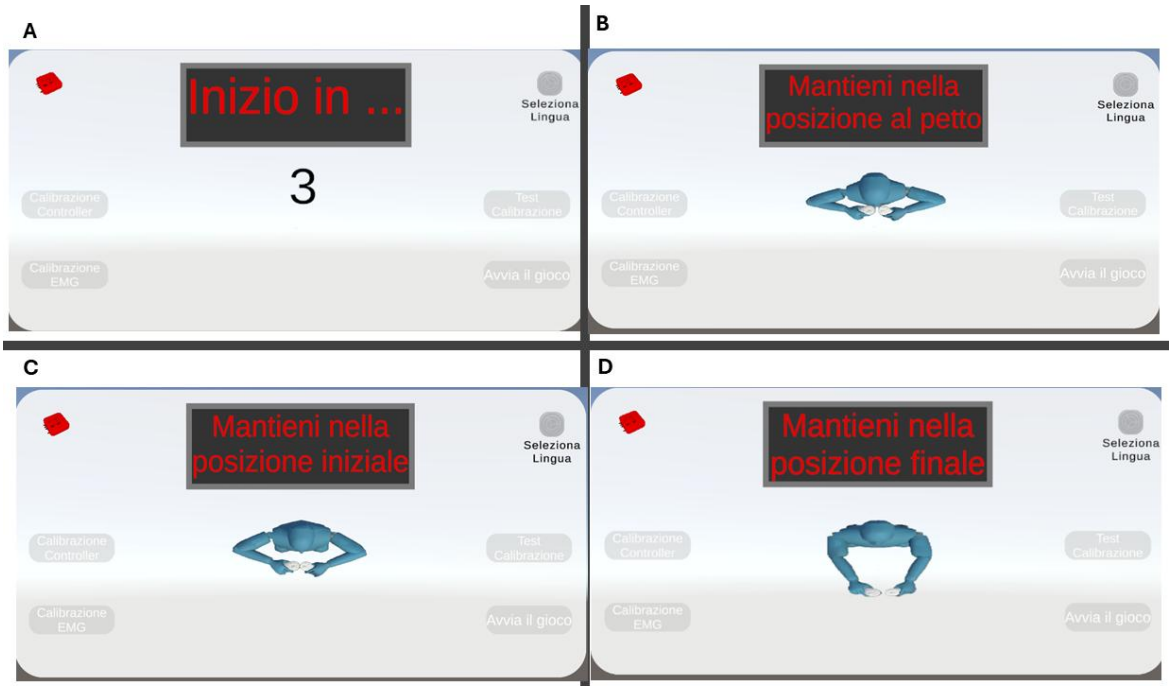

Figura 23: CalibrationView nelle varie fasi della calibrazione del ROM . A) Countdown. B) Posizione al petto. C) Posizione iniziale. D) Posizione finale.

Per ciascuna delle tre posizioni è stato implementato un ciclo *while* della durata di sei secondi. La misurazione del tempo è ottenuta utilizzando la variabile *Time.deltaTime*, fornita nativamente dal motore Unity, che restituisce l'intervallo temporale trascorso tra un *frame* e il successivo. Ad ogni iterazione del ciclo vengono campionate e accumulate le coordinate *z* e il tempo trascorso dall'inizio dell'acquisizione. Al termine, i vettori raccolti sono trasferiti al *CalibrationData* i tramite il metodo *CalibrationData.Instance.AddZData()*. Completata la terza posa, il sistema notifica all'utente il completamento della calibrazione del ROM tramite feedback testuale, aggiorna il *flag isControllerCalibrated* e invoca *CalibrationData.Instance.ComputeZStats()* per il calcolo della mediana e deviazione standard dei dati raccolti nelle tre posizioni. Le immagini mostrate nell' interfaccia utente durante questa fase hanno associato il componente *custom ImageFader.cs*. Attraverso due *coroutine* dedicate, *FadeIn()* e *FadeOut()*, l'immagine viene progressivamente resa visibile o nascosta mediante la variazione temporale del valore di trasparenza. Il componente disattiva inoltre il *raycast* per evitare interferenze con l'interazione dell'utente e, al termine dell'animazione di dissolvenza in uscita, provvede a disattivare automaticamente l'oggetto grafico associato.

Acquisizione della MVC

Terminata la calibrazione, se la sessione prevede l'utilizzo dell'EMG, viene abilitato il pulsante "CALIBRAZIONE EMG" e disabilitati gli altri (Figura 24 A). Cliccando su di esso viene mostrato un *pop-up* informativo in maniera analoga alla calibrazione del ROM Figura 24 B.

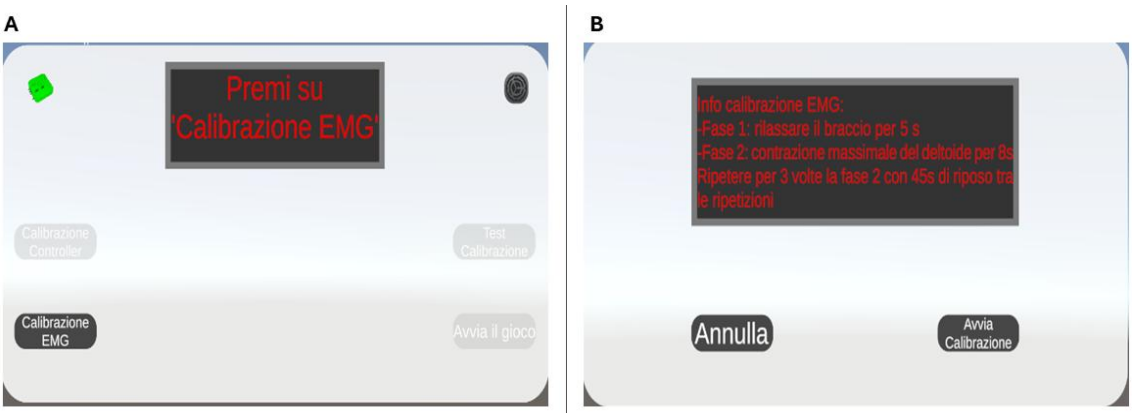

Figura 24: Interfacce per l'acquisizione MVC. A) CalibrationView durante la fase di Calibrazione EMG. B) Pop-up Calibrazione EMG.

Proseguendo il *Manager* avvia la *coroutine StartCalibrationEMG(),* che predispone l'interfaccia, abilita la vista informativa sul flusso UDP quando necessario e introduce un preavviso con countdown tramite *StartWithDelay()* (Figura 25 A). La logica operativa risiede nella *coroutine CalibrateEMG(),* dove è innanzitutto verificata la disponibilità di una connessione valida con la *SyncStation* e in caso positivo avvia l'acquisizione attraverso il metodo *StartAcquisition*() dell'istanza di *SyncStationConnection*. Dopo un breve intervallo di sicurezza, i dati nel *buffer* vengono azzerati tramite il metodo *custom ClearAllList()* del *DuePlusDataContainer*, che esegue il metodo *built-in Clear()* di tutte le liste presenti all'interno della classe. Successivamente si procede alla fase di misurazione del rumore di base, durante la quale l'utente è invitato a rilassare il braccio (Figura 25 B) e i dati acquisiti presenti nel *buffer DuePlusDataContainer* sono trasferiti al *CalibrationData* attraverso il metodo *CalibrationData.Instance.AddNoiseData().* In seguito, vengono eseguite le tre ripetizioni per acquisire la contrazione massimale del muscolo target, ciascuna introdotta da un feedback testuale e guidata da un *countdown*, utilizzando nuovamente la *coroutine StartWithDelay* (Figura 25 C). Per ogni ripetizione si registrano indice di inizio dei dati grezzi e dell'inviluppo (*startIndexRowEMG* e *startIndexEnvEMG*) così da trasferire al *CalibrationData*, con il metodo *CalibrationData.Instance.AddEMGData()*, i segmenti pertinenti alla ripetizione. Tra una ripetizione e la successiva è prevista una fase di riposo per evitare fenomeni di fatica, con temporizzazione e feedback testuale mostrati nella UI (Figura 25 D).

Figura 25:CalibrationView nelle varie fasi dell'acquisizione dell'MVC: A) Countdown. B) Acquisizione del rumore di fondo (muscolo a riposo). C)Esecuzione della contrazione massimale. D) Fase di recupero.

Al termine del protocollo, il sistema produce un feedback di completamento, calcola le statistiche riassuntive con *CalibrationData.Instance.ComputeEMGStats()* e imposta a *true isEMGCalibrated*. Opzionalmente la procedura implementa con la stessa logica anche l'acquisizione della MVC del bicipite e tricipite nel caso in cui siano collegate due sonde EMG. Il calcolo dell'MVC viene eseguito a partire dall' inviluppo dell'EMG ed è implementato nel metodo *CalcoloMVC()* di *CalibrationData*. L'MVC finale è ottenuto calcolando il massimo dell'inviluppo del segnale EMG nelle tre ripetizioni.

Test Calibrazione

Se la stima dell'MVC non è prevista o è stata ultimata nella *CalibrationView* viene abilitato il pulsante "TEST CALIBRAZIONE". In risposta all'interazione dell'utente il *CalibrationManager* avvia la fase di test mediante il metodo *ActivateTestManager()*. Tale metodo verifica preliminarmente che tutte le fasi precedenti della calibrazione siano state completate e, qualora previsto dal tipo di sessione, abilita gli elementi necessari alla valutazione delle attivazioni muscolari (script dedicati ed elementi grafici). Successivamente attiva il *TestManager* e aggiorna l'interfaccia utente con la vista specifica per il test (TestView, Figura 26).

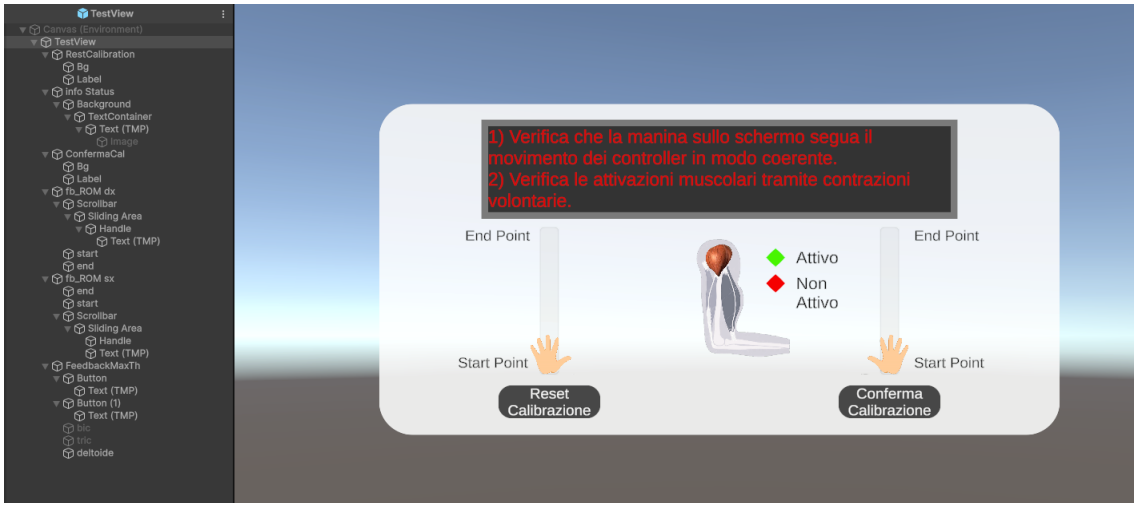

Figura 26: TestView e relativi GO.

Il *TestManager* implementa la logica operativa della verifica della calibrazione effettuata tramite due classi distinte: *TestControllerCalibration* e *TestCalibrationEMG*. La classe *TestControllerCalibration* è dedicata alla valutazione dei punti di inizio e fine del movimento riabilitativo e consente la visualizzazione in tempo reale dello spostamento del controller lungo l'asse *z*. Nel mondo virtuale tale movimento è rappresentato dall'avanzamento dell'immagine di una mano su una *Scrollbar* la cui implementazione e logica sarà discussa successivamente (cfr. sez. Biofeedback cinematici). In caso di dati incoerenti, l'interfaccia segnala l'errore rendendo permanente la variazione cromatica dell'indicatore, che diventa rosso e rimane statico. Se invece il controller oltrepassa temporaneamente i limiti ammessi, la variazione del colore viene resa momentanea rossa tramite una *coroutine* dedicata, fornendo un feedback immediato ma non bloccante all'utente.

La classe *TestCalibrationEMG* è responsabile della verifica delle attivazioni muscolari tramite un metodo a doppia soglia analogo a quello utilizzato nel biofeedback EMG (cfr. sez. "Biofeedback EMG"). All'attivazione del GO, il componente si assicura che la catena di acquisizione sia disponibile e, in caso contrario, inizializza la connessione e avvia l'acquisizione tramite l'istanza di *SyncStationConnection*. Lo stato dell'attività muscolare rilevata è comunicato all'utente attraverso la variazione cromatica dell'indicatore grafico associato al muscolo monitorato: verde in caso di attivazione stabile, rosso in caso contrario e grigio quando il numero di campioni disponibili non è sufficiente a produrre una valutazione attendibile.

Terminata la verifica, se l'utente decide di ripetere la calibrazione attraverso il pulsante "RESET CALIBRAZIONE", il *CalibrationManger* esegue la *coroutine ResetCalibration(),* che ripristina tutti i dati della calibrazione chiamando il metodo *ClearAllCalibrationData()* dell'istanza del *CalibrationData* e riporta la UI e i *flag* allo stato iniziale. In caso contrario, se la calibrazione viene confermata tramite il pulsante "CONFERMA CALIBRAZIONE", viene eseguito il metodo *ConfirmCalibration()* del *CalibrationManager*, il quale chiude il test, salva i dati e aggiorna gli indicatori per permettere l'accesso dell'utente alla fase successiva.

Avvio dell'exergame

Al termine del test l'utente visualizza nuovamente la *ClibrationView,* nella quale il pulsante "TEST CALIBRAZIONE" non viene disabilitato, così da consentire eventuali verifiche aggiuntive. Contestualmente viene reso disponibili all'utente il pulsante "AVVIA GIOCO" per l'avvio dell'*exergame*. Se l'utente seleziona l'avvio del gioco, il *CalibrationManager* attiva la *coroutine* responsabile del caricamento asincrono della scena successiva, tramite il metodo *custom LoadNextScene()*. Tale procedura esegue una nuova verifica dello stato della calibrazione tramite i *flag* dedicati e, in caso di incongruenze, notifica all'utente l'errore impedendo la prosecuzione. Qualora tutte le condizioni risultino soddisfatte, la *coroutine* avvia il caricamento della scena successiva tramite il metodo *built-in LoadSceneAsync()* del *SceneManager* e rimane in attesa del completamento monitorando, ad ogni *frame*, l'avanzamento dell'operazione attraverso la proprietà *isDone* del risultato dell'operazione asincrona (AsyncOperation). Una volta ultimato il caricamento, l'utente viene trasferito alla scena successiva, nella quale avviene la connessione al server e l'inizializzazione necessaria all'avvio *dell'exergame*.

Task motorio e logica di controllo dei remi

Il protocollo riabilitativo proposto prevede l'esecuzione di un task motorio di *reaching* planare dell'arto superiore eseguito sul piano orizzontale. Nello specifico, il gesto richiesto consiste in un movimento di flesso-estensione coordinata del gomito e della spalla, in cui la mano partendo da una posizione centrale, descrive una traiettoria rettilinea in direzione antero-posteriore fino alla massima estensione dell'arto.

Tale cinematica lineare viene traslata all'interno dell'ambiente virtuale attraverso un algoritmo di *mapping* che converte il movimento dell'utente nella simulazione circolare di una remata. La corrispondenza tra il gesto fisico (spazio reale) e l'animazione dell'avatar (spazio virtuale) è suddivisa in due fasi distinte, funzionali alla meccanica del canottaggio:

1. **Fase attiva (propulsiva):** Il movimento di estensione del gomito nel mondo reale (definito movimento di "andata"), che copre un *Range of Motion* (ROM) di circa 90° dalla posizione iniziale alla completa estensione, corrisponde alla fase efficace della remata. In questa fase, definita tecnicamente "di attacco", il remo virtuale risulta immerso in acqua e la pala esercita la spinta necessaria alla propulsione dell'imbarcazione. Nel mondo virtuale, questo input lineare viene trasformato in una rotazione di 180°.

2. **Fase passiva (recupero):** Il conseguente movimento di flessione del gomito (movimento di "ritorno"), necessario per riportare l'arto dalla massima estensione alla posizione di partenza, corrisponde alla fase non efficace del ciclo di voga. In questo frangente, la pala virtuale si solleva dall'acqua ed esegue un movimento di ripresa aerea (rotazione di 180°), ritornando alla posizione iniziale senza contribuire attivamente al moto della barca.

Inoltre, come anticipato nel paragrafo precedente, il *mapping* è progettato per adattarsi al ROM residuo del soggetto garantendo che ogni partecipante (rematore virtuale) possa fornire lo stesso contributo al compito collaborativo, indipendentemente dalle capacità residue.

Per ottenere questo effetto sono stati implementati due script *CircularMovements.cs* e *RowManager.cs*.

Mapping da movimento lineare a movimento circolare

La classe *CircularMovements* ha il compito di mappare gli spostamenti lineari del controller relativi alla direzione antero-posteriore, ovvero lungo l'asse Z nel sistema di riferimento virtuale, in una traiettoria angolare appartenente ad una circonferenza definita sul piano YZ. Le coordinate spaziali derivanti da tale mappatura angolare determinano la posizione di un Game Object (*GO*), il quale funge da riferimento per l'orientamento e la rotazione del remo virtuale. All'interno dell'ambiente Unity sono state implementate due istanze distinte di questa classe, ognuna associata ad un proprio GO e referenziate rispettivamente ai controller destro e sinistro, permettendo di governare contemporaneamente i due remi con la stessa logica. Tuttavia, poiché *l'exergame* finale prevede l'utilizzo di un singolo remo, il sistema in fase di esecuzione, abilita selettivamente l'istanza e il GO corrispondenti alla lateralità selezionata dall'utente, disabilitando le componenti ridondanti. I GO impiegati sono definiti come *Empty*, in quanto contengono esclusivamente la componente *Transform* necessaria alla gestione delle coordinate e

dell'orientamento, risultando privi di componenti di rendering e, pertanto, invisibili all'utente durante la sessione riabilitativa.

Per quanto concerne la logica operativa, lo script in fase di inizializzazione ottiene i valori di calibrazione $Z_{\text{START}}, Z_{\text{END}}$ relativi al punto iniziale e finale del movimento riabilitativo per il lato destro o sinistro dal *CalibrationData*. Al fine di attenuare micro-tremori e piccole variazioni, è stato implementato un filtro a media mobile basato su un *buffer First-In-First-Out (FIFO)* a 5 campioni, inizializzato con il valore di Z_{START}. Ad ogni *frame* (*Update*) la posizione locale del controller lungo l'asse Z viene utilizzata per aggiornare il *buffer*. Successivamente viene calcolata la media dei valori presenti al suo interno ($\bar{z}(t)$), utilizzata nella logica di conversione circolare, richiamando il metodo *CircularMovement(). Tale procedura* necessita della stima della direzione del movimento del controller per eseguire correttamente il *mapping* tra le fasi del movimento riabilitativo e quelle della remata. In particolare, il metodo è caratterizzato da

- **Verifica della soglia minima per lo spostamento**:

$$|\Delta z| = |\bar{z}(t) - \bar{z}(t - \Delta t)| > MDA$$

  Con il parametro MDA (*MinDistanceAllowed)* impostato a 0.001
  Se questa condizione non è rispettata viene eseguito il *return* del metodo e la posizione del GO non viene aggiornata.

- **Determinazione della direzione iniziale:** nella fase iniziale non viene impostata una direzione a priori ma viene analizzato l'andamento della coordinata Z in un breve intervallo temporale (10 *Update*). Se quest'ultimo risulta crescente allora si assume che la direzione iniziale sia quella in avanti, in caso sia decrescente allora indietro. La direzione è impostata mediante apposito *flag* (*isMovingForward*)
- **Monitoraggio del cambio di direzione del controller**: la logica per rilevare i cambiamenti di direzione del controller si basa sul segno di Δz(t), sulla direzione del controller all' istante precedente, rappresentata dal booleano $IsMovingFoward$ (*true* avanti, *false* indietro) e sul superamento di una certa soglia *MinDistanceToChangeDirection* (MDCD) impostata a 0.02:
  1. Indietro → avanti

$$\bar{z}(t) > \bar{z}(t - \Delta t) \ \& \ |\Delta z| > MDCD \ \& \ \sim \text{IsMovingFoward}$$

  2. Avanti → indietro

$$\bar{z}(t) < \bar{z}(t - \Delta t) \ \& \ |\Delta z| > MDCD \ \& \ \text{IsMovingFoward}$$

  3. Nessuna variazione

$$(\bar{z}(t) \geq \bar{z}(t - \Delta t) \ \& \ \text{IsMovingFoward}\ ) \mid \ (\bar{\bar{z}}(t) \leq \bar{z}(t - \Delta t) \ \& \ \sim\text{IsMovingFoward}\ )$$

- **Calcolo della posizione angolare:** Il valore $\bar{z}(t)$ viene limitato nell'intervallo $[Z_{\text{START}}, Z_{\text{END}}]$ e normalizzato, attraverso la combinazione delle funzioni *Clamp e InverseLerp* entrambe appartenenti alla classe *Mathf* di Unity.
  Il *mapping* circolare avviene in maniera differente a seconda della direzione del controller (Figura 27). Quando il soggetto muove il controller dal punto iniziale a quello finale, il valore normalizzato della coordinata *z* va da 0 a 1. In questa fase, l'angolo in radianti viene calcolato tramite interpolazione dalla funzione *Mathf.Lerp*(π, 0, t), con valori che vanno da π a 0. Nella fase passiva del movimento, ovvero dal punto finale a quello iniziale, il valore normalizzato va da 1 a 0, mentre l'angolo è calcolato -*Mathf.Lerp*(π, 0, t), con valori che variano da 0 a -π.

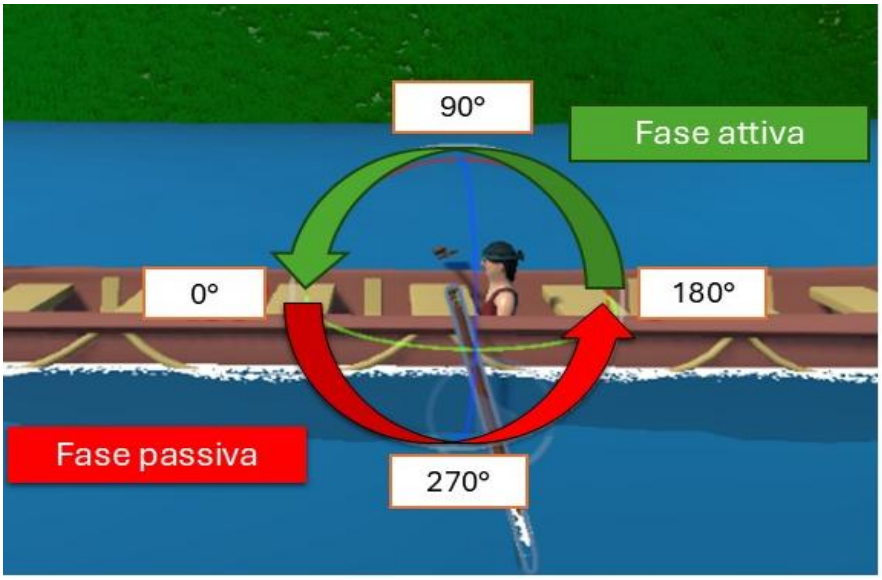

Figura 27: Rappresentazione della traiettoria circolare descritta dal mapping angolare. In verde la traiettoria circolare corrispondente alla fase di estensione (fase attiva), in rosso la semicirconferenza descritta durante la fase di flessione (fase passiva). La traiettoria descritta non è quella del remo, ma del GO che lo controlla.

- **Aggiornamento della posizione del *Game Object:*** In caso di variazione della direzione viene controllato se quest'ultima avviene in corrispondenza dei limiti ottenuti durante la calibrazione. In caso positivo o se non avviene nessuna variazione l'angolo calcolato (θ(t)) viene utilizzato dal metodo *UpdatePosition()* per aggiornare la posizione tridimensionale del GO:

$$\mathrm{p(t)} = \begin{bmatrix} x \\ r\sin\theta(\mathrm{t}) \\ r\cos\ \theta(\mathrm{t}) \end{bmatrix}$$

  In caso contrario la posizione angolare non viene aggiornata, ma viene richiamata una coroutine per muovere linearmente il GO nella nuova posizione angolare. Questo approccio risulta necessario per evitare spostamenti istantanei del remo in quanto la

posizione angolare, per come è stato implementato il *mapping*, passa immediatamente da θ a $-\theta$ o viceversa.

- **Movimento lineare:** in caso di movimenti incompleti il remo si muove linearmente mediante la coroutine *MoveBetweenPositionsLinear*. La procedura prende in ingresso l'angolo di partenza (θ_1) e quello di arrivo (θ_2), il raggio r e la durata della transizione. Inizialmente stima le due posizioni nello spazio:

$$\mathrm{p1} = \begin{bmatrix} x \\ r \sin\ \theta_1 \\ r \cos\ \theta_1 \end{bmatrix}$$

$$\mathrm{p2} = \begin{bmatrix} x \\ r \sin\ \theta_2 \\ r \cos\ \theta_2 \end{bmatrix}$$

Ad ogni *Update* esegue un'interpolazione tridimensionale basata sul tempo trascorso dalla chiamata Δt, misurato tramite il metodo *Time.deltaTime* di Unity, e la durata della transizione T impostata a 0.2 secondi: $Vector3.Lerp(\mathrm{p1}, \mathrm{p2}, \tau)$ con $\tau = {}^{\Delta t}/_{T}$ Il valore dell'interpolazione viene utilizzato per aggiornare la posizione del GO ad ogni *frame*.

Nell'applicazione il raggio della circonferenza è stato impostato a 0.75 e alla posizione del GO viene sommato un offset di 1.2 lungo l'asse Y, coerentemente con l'offset applicato dal sistema alla vista del soggetto quando è posizionato sulla barca. In questo modo la circonferenza è centrata sull'avatar virtuale che governa i remi, permettendo al remo virtuale di seguire una cinematica simile al gesto di vogata reale.

Logica di controllo del remo

Lo script *RowManager* coordina la cinematica del remo e la trasmissione della forza propulsiva da trasferire alla barca.

All'interno della classe ad ogni *FixedUpdate*, il metodo *RotatePivot*() calcola il vettore direzione che unisce il fulcro del remo alla posizione corrente del GO pilotato dalla classe *CircularMovements*. Tale vettore viene utilizzato per definire la rotazione da applicare al pivot, tramite la funzione *built-in* di Unity *Quaternion.LookRotation*(), per allineare l'asse longitudinale del remo con la posizione del GO calcolata da *CircularMovements*.

Il calcolo della forza propulsiva generata dal remo è gestito dal metodo *CalculateSpeed*(), che opera esclusivamente quando il remo risulta immerso nel fluido. Il sistema misura lo spostamento lineare lungo l'asse Z del controller, calcolando la differenza di posizione rispetto al *frame* precedente. La velocità istantanea è ottenuta come rapporto tra questo incremento spaziale e il tempo fisso di aggiornamento del motore fisico (*Time.fixedDeltaTime,* 0.02 *s*). Per evitare

aggiornamenti dovuti a micromovimenti, è stata implementata una soglia di spostamento (*distanceThreshold*) pari a 0.01.

La velocità così ottenuta è utilizzata dal metodo *Move*(), che delega alla classe *ShipNetwork.MoveShip*() l'aggiornamento delle componenti di spinta in avanti e di coppia per la rotazione della barca. Tali forze sono modulate dai parametri *forwardSpeed* e *torqueSpeed* impostate in fase di sviluppo rispettivamente a 0.75 e 0.08 per garantire una spinta coerente. Se il remo non è immerso, la velocità applicata è nulla, riproducendo la naturale alternanza tra fase propulsiva e fase di recupero.

La verifica dello stato di immersione del remo avviene mediante la gestione di eventi *trigger* gestiti nativamente da Unity: *OnTriggerEnter* e *OnTriggerExit*. Questi eventi vengono invocati automaticamente quando il *collider* associato al remo entra o esce dal volume che definisce la superficie dell'acqua, attivando o disattivando una variabile booleana.

In un tipico flusso operativo, *CircularMovements* trasforma la traiettoria lineare del controller eseguita dall'utente durante il movimento riabilitativo in una traiettoria circolare coerente con la cinematica della remata. *RowManager* utilizza tale posizione per orientare il pivot del remo e traduce lo spostamento lungo Z in una forza per il modello fisico della barca, assicurando un'interazione realistica tra l'input dell'utente e il comportamento del remo nel mondo virtuale.

Progettazione e implementazione dei Biofeedback

I biofeedback sviluppati possono essere divisi in due categorie a seconda della funzione che devono misurare:

1. **Basati su variabili cinematiche**: orientati a valutare la qualità del gesto riabilitativo. Il *reaching* deve essere completo e ritmico, ovvero eseguito sull' intero ROM residuo del con movimenti fluidi e continui. Tali parametri sono stimati attraverso l'analisi della posizione e della velocità del controller VR impugnato dal paziente durante l'esecuzione del task motorio.

2. **Basati sull'attività muscolare**: finalizzati a monitorare e informare il paziente sulla correttezza del movimento valutata in termini di assenza di sovra-attivazioni muscolari del deltoide laterale, riconducibili a strategie compensatorie di abdo-adduzione della spalla durante il compito motorio.

Nell'ambiente virtuale, gli indicatori grafici associati ai biofeedback sono stati implementati come elementi bidimensionali (2D) dinamici, collocati all'interno di un Canvas, posizionati nello spazio 3D a una distanza di 0.5 metri dal punto di osservazione dell'avatar, con un'inclinazione di 70 gradi rispetto all'asse verticale (Figura 28). Questa configurazione è stata scelta per allineare il piano visivo del feedback con il piano di movimento reale dell'utente (antero-posteriore). Un posizionamento verticale del Canvas avrebbe indotto una dissociazione tra il movimento del braccio (avanti-indietro) e il feedback visivo (alto-basso), inoltre il posizionamento orizzontale garantisce una configurazione meno invasiva permettendo al soggetto di esplorare con più facilità l'ambiente circostante.

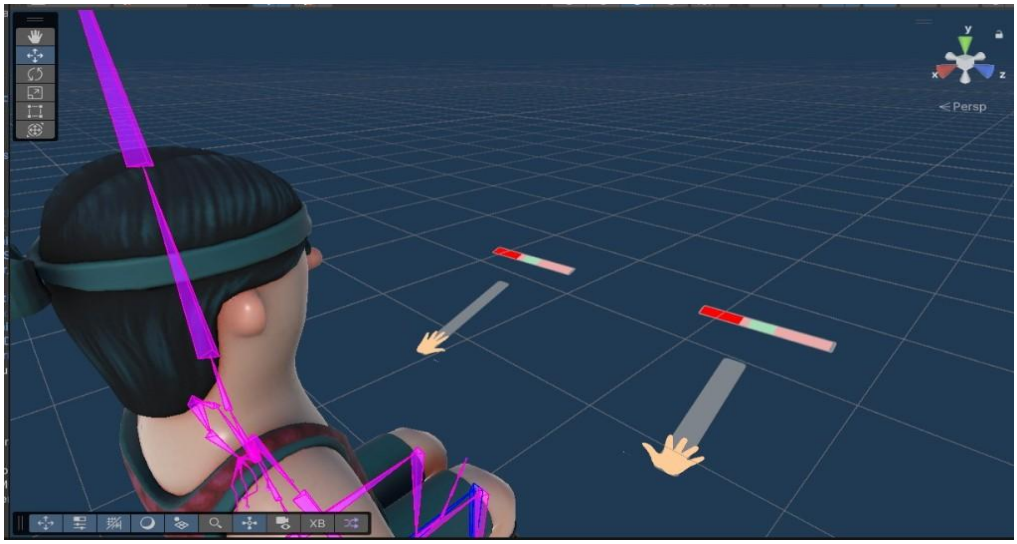

Figura 28: Posizionamento del Canvas contenete gli elementi grafici relativi ai biofeedback rispetto all'avatar virtuale.

L'architettura software dedicata alla gestione del biofeedback segue una logica gerarchica e modulare, concepita per coordinare le risorse locali del giocatore con le entità sincronizzate in rete. All'interno della gerarchia di Unity, il *Canvas* che ospita le interfacce è posizionato come

oggetto "figlio" del *VR Player*, tuttavia, durante le fasi iniziali e di calibrazione, tutti i suoi componenti grafici e gli script di controllo rimangono disabilitati.

L'attivazione del sistema avviene nel momento in cui l'utente conclude la calibrazione ed entra ufficialmente nella sessione di gioco. Questo passaggio è mediato dal *VRPlayerNetwork* (l'entità creata per gestire il giocatore nel contesto *multiplayer*), il quale, non appena viene istanziato dal server, comunica al *VR Player* (entità locale*)* la posizione assegnata sulla barca (remo destro o sinistro) e gli fornisce il riferimento dinamico al relativo *RowManager.* Quest'ultimo passaggio è critico, in quanto il remo è un'entità gestita dal sistema di *networking* e non appartiene alla gerarchia locale del *VR Player*.

Il *VR Player* riceve queste informazioni e delega l'inizializzazione effettiva richiamando il metodo *custom SetUpBiofeedback()*. Questa funzione agisce come un ponte verso il *FeedbackManager*, il quale avvia la procedura di configurazione dell'interfaccia:

1. **Attivazione e visualizzazione**: Il manager abilita il Game Object principale del rendendo visibile il Canvas nel campo visivo del paziente.
2. **Selezione della lateralità**: In base al parametro *IsRight*, il sistema configura la visualizzazione corretta. Le interfacce grafiche sono infatti sdoppiate per gestire i due lati: il *manager* provvede a disabilitare i componenti relativi al lato non utilizzato (sia a livello di UI che di logica) e ad attivare esclusivamente quelli pertinenti alla lateralità corrente.
3. **Trasmissione delle dipendenze**: Il *FeedbackManager* distribuisce i riferimenti del *RowManager* (proveniente dalla rete) e *CircularMovements* (locale al *VR Player*), ai vari sotto moduli garantendo che ogni sistema di biofeedback operi sulle istanze corrette durante la sessione.

Biofeedback cinematici

Per quanto riguarda i biofeedback cinematici, la progettazione si è focalizzata sulla creazione di indicatori visivi di facile comprensione, che potessero fornire informazioni sia in termini di *Knowledge of Performance* che di *Knowledge of Results*.

L'indicatore dedicato al monitoraggio del Range of Motion (ROM) e della completezza del gesto è mostrato in Figura 29 e consiste in una mano virtuale che rappresenta i movimenti del controller lungo direzione antero-posteriore. Quest'ultima si muove all'interno di una barra rappresentate il ROM residuo del soggetto ottenuto durante la fase di calibrazione.

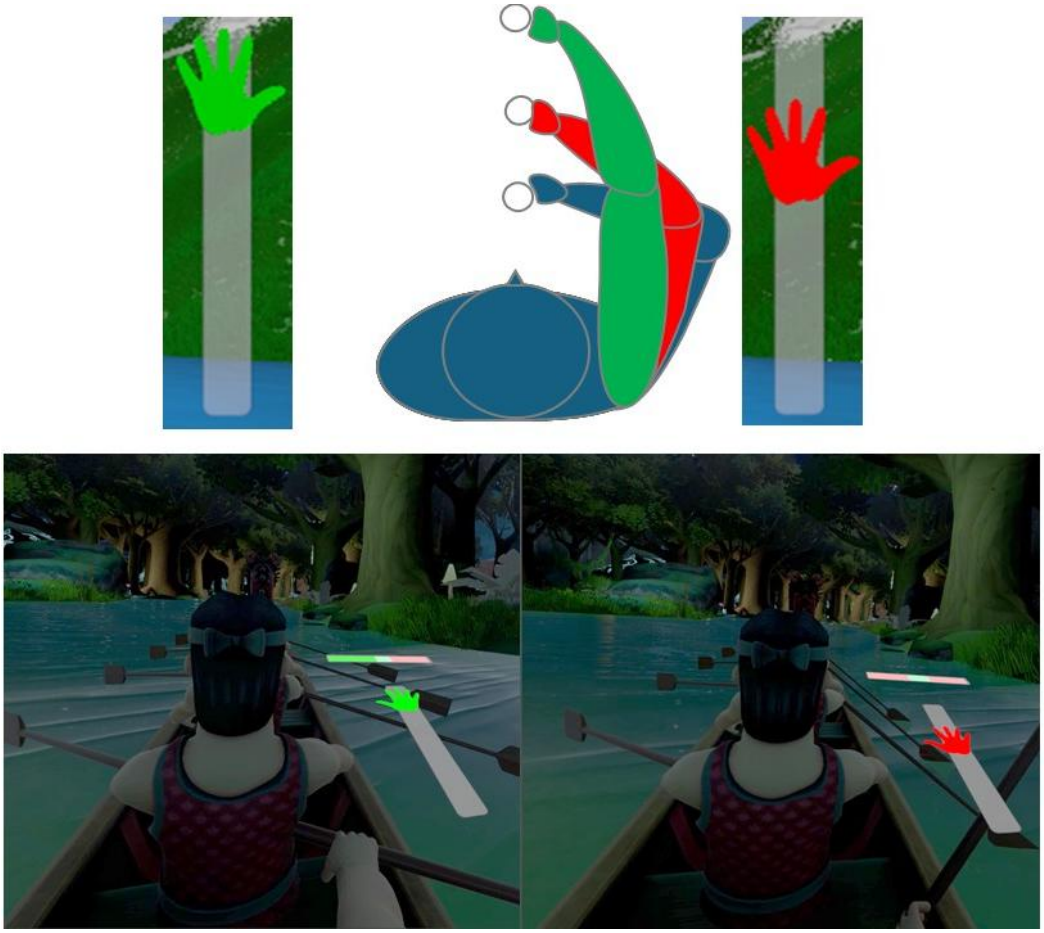

Figura 29: Rappresentazione del biofeedback associato al ROM nei casi di movimento completo o incompleto.L'immagine in alto schematizza un soggetto che esegue un movimento completo, dal punto iniziale (braccio blu) al punto finale (braccio verde, biofeedback a sinistra) e incompleto (braccio rosso. biofeedback a destra) con lateralmente i biofeedback associati. In basso a sinistra è mostrato un frame dell'exergame VR con il biofeedback del movimento completo, mentre a destra per un movimento incompleto.

Dal punto di vista grafico, tale indicatore è stato realizzato attraverso la personalizzazione dell'elemento *Scrollbar* nativo di Unity utilizzato tipicamente nelle interfacce utente. Questo Game Object è costituito da un'area di scorrimento (*Sliding Area*) e da un cursore mobile, *l'Handle*, che nel progetto è stato sostituito dall'icona di una mano per favorire l'interpretazione del paziente. Normalmente la *Scrollbar* è un elemento con il quale l'utente può interagire selezionando e spostando l'*Handle* lungo la *Sliding Area*. Tuttavia, tale funzionalità è stata rimossa implementando una logica che governa lo spostamento dell'*'Handle* agendo direttamente sul parametro *value*. Quest'ultimo determina la posizione dell'*Handle* lungo l'area di scorrimento, con il valore riportato come frazione tra le estremità, ovvero il valore 0 rappresenta l'estremità in basso, mentre 1 all'estremità in alto.

L'algoritmo per lo spostamento dell'*Handle* è implementato dalla classe *custom ROMbiofeedback* e utilizza la stessa logica di *CircularMovements*:

- Il valore della posizione lungo l'asse z del GO relativo al controller viene mediato con gli ultimi 5 valori registrati.
- Se viene registrato uno spostamento significativo rispetto alla posizione precedente viene aggiornato il parametro *value* della *Scrollbar* utilizzando il valore della coordinata z corrente normalizzato con gli estremi del ROM. La normalizzazione viene effettuata attraverso il metodo *InverseLerp* della libreria Mathf, che implementa la formula del *min-max scaling* prendendo in ingresso i due estremi, Z_{START} e Z_{END}, ottenuti dal *CalibrationData* e la coordinata z corrente (Z):

$$n = \frac{Z - Z_{START}}{Z_{END} - Z_{START}} \tag{4}$$

  Tale operazione garantisce che l'intera estensione della barra grafica corrisponda esattamente alle reali capacità motorie residue del soggetto.
- Infine, viene utilizzata la logica per rilevare i cambiamenti di direzione del controller al fine di verificare che quest'ultimi avvengano in corrispondenza dei punti acquisiti durante la calibrazione. Se questo non avviene, quindi se il paziente effettua un movimento incompleto, fermandosi prima del punto finale o non rientrando in posizione, viene richiamata una *coroutine* che modifica temporaneamente il campo *Color* dell'immagine della mano, che da neutro (rosa) diventa rosso per 0.5 secondi. In caso contrario, quindi se il soggetto esegue correttamente il movimento, la mano diventa temporaneamente verde.

Il biofeedback così progettato ha un duplice ruolo, da un lato informa il paziente sulla performance (*KoP)* che sta eseguendo, ovvero su che percentuale del ROM residuo sta eseguendo il movimento, tramite una visualizzazione indiretta (posizione relativa della mano lungo la barra) dall'altro informa esplicitamente il soggetto sulla correttezza del movimento (KoR) tramite variazione cromatica dell'indicatore. Inoltre, la rappresentazione dei movimenti della mano dell'utente nell'ambiente virtuale fornisce un'indicazione implicita delle ritmicità del gesto riabilitativo.

In merito alla coordinazione temporale del task, il sistema adotta una strategia di biofeedback multimodale volta a favorire la sincronizzazione di fase e la regolarità cinematica tra i partecipanti senza ricorrere a un indice di valutazione esplicito. La sincronizzazione viene mediata primariamente da un segnale acustico ritmico che, riproducendo il suono di un tamburo, scandisce gli istanti iniziali e finali di ogni ciclo di movimento, fornendo un riferimento uditivo condiviso per l'allineamento dei ritmi motori. Questa tipologia di feedback riduce il carico cognitivo dei partecipanti, il suono indica il momento esatto di "attacco" e "ripresa" della vogata, riducendo la complessità di doversi coordinare guardando i movimenti degli altri avatar.

Parallelamente, il monitoraggio della velocità del controller assume un ruolo centrale nel protocollo riabilitativo, poiché tale parametro determina direttamente la forza propulsiva impressa al remo e la conseguente avanzata della barca all'interno dell'ambiente virtuale. In uno scenario collaborativo ideale, i soggetti dovrebbero contribuire in egual misura al moto della barca per assicurare un moto fluido e coordinato. Al fine di raggiungere questo scopo il sistema impone una velocità target alla quale i soggetti devono sincronizzarsi. Il biofeedback visivo è dunque progettato per informare costantemente l'utente sulla propria prestazione (*performance*) cinematica, evidenziando tre condizioni distinte: una situazione di *under performance*, qualora la velocità risulti inferiore a quella desiderata, una di *over performance*, nel caso di movimenti eccessivamente rapidi e infine la condizione ottimale (*correct performance)*, in cui la cinematica rispetta i vincoli imposti dal sistema.

Questa informazione viene restituita in tempo reale attraverso una barra orizzontale che modifica la propria estensione spaziale in funzione alla velocità misurata e segnala la correttezza del gesto cambiando la propria colorazione: verde per la *correct performance,* rosso per *under performance* o *over performance*. Quest'ultima si muove all'interno di un contenitore di colore grigio, il quale funge da cornice per uno sfondo rosso, deputato alla segnalazione degli stati di errore, su cui è sovrapposta una zona centrale di colore verde che rappresenta il range di tolleranza sulla velocità target. Per facilitare la lettura della sovrapposizione tra l'indicatore e le zone target sottostanti, è stata applicata una semitrasparenza tramite il canale *alpha* della proprietà colore; questo accorgimento permette al paziente di percepire simultaneamente la propria prestazione attuale e i limiti della zona di successo, favorendo una correzione intuitiva del gesto motorio. In Figura 30 è mostrata la rappresentazione grafica della barra nelle tre condizioni citate precedentemente.

Figura 30: Rappresentazione grafica del biofeedback sulla velocità nel caso di correct performance *al centro: la velocità di movimento del soggetto rispecchia i limiti imposti dal sistema, la barra diventa verde e ricade dentro la zona target; under performance a sinistra: la velocità è inferiore rispetto al target, la barra non raggiunge l'obiettivo e divento rossa; over performance a destra : la velocità è maggiore del limite superiore imposto dal sistema, la barra super il target e diventa rossa. Il livello massimo di riempimento della barra rappresenta la saturazione del feedback (2 volte la velocità target).*

L'architettura software del biofeedback si basa sui componenti *Image* del framework Unity UI. Durant l'implementazione, non è stata assegnata la proprietà *Source Image*, permettendo di gestire ogni elemento come un rettangolo di colore pieno, la cui estetica dipende esclusivamente dalla proprietà *Color*. La dimensione e la posizione spaziale all'interno del Canvas sono determinate dai parametri del *RectTransform*, tra cui:

- *Anchors*: definiscono la relazione spaziale tra l'oggetto UI e il suo contenitore genitore permettendo alla barra del biofeedback di mantenere le proporzioni corrette indipendentemente dalla dimensione del Canvas.
- *Pivot:* rappresenta il punto centrale attorno al quale avvengono tutte le trasformazioni, come la rotazione e la scalatura. La sua corretta impostazione è cruciale per la dinamica del feedback: un perno posizionato lateralmente permette alla barra di espandersi da un lato verso l'altro, mentre un perno centrale abilita un riempimento simmetrico, fondamentale per la visualizzazione della zona target.
- *SizeDelta*: esprime la dimensione del rettangolo in relazione alla distanza tra gli ancoraggi. Questa proprietà è lo strumento principale per la manipolazione via software, poiché consente di variare la larghezza o l'altezza dell'immagine in unità di pixel (o unità Canvas) in risposta agli input cinematici.

La logica che governa il biofeedback è implementata nella classe *ContributeFeedbackManager*, la quale necessita di un elemento esterno (il *FeedbackManager*) per essere inizializzata tramite l'apposito metodo *custom Initialize()*. Per la corretta inizializzazione la procedura necessita in ingresso la lateralità corrente espressa tramite un parametro *booleano* (*true* per destra e *false* per sinistra). Durante questa fase, il sistema inizializza le variabili e abilita gli elementi del biofeedback coerentemente con il lato selezionato. Inoltre, provvede al setup geometrico della zona target attraverso la manipolazione delle proprietà del *RectTransform*, impostando gli ancoraggi e il perno al centro della barra contenitore, e definendo l'ampiezza (componente *x* di *SizeDelta*) come percentuale della lunghezza totale del contenitore. Questa percentuale, impostata al 20%, rappresenta la tolleranza del biofeedback rispetto alla velocità target (parametro *targetSpeed*), garantendo un riscontro positivo nel range [0.8 * *targetSpeed; 1.2 * targetSpeed].* L'adozione di un margine di tolleranza è necessaria per compensare la naturale variabilità del movimento umano ed eventuale rumore del sistema VR, che renderebbe quasi impossibile sincronizzarsi ad un valore puntuale di velocità generando fenomeni di frustrazione nel soggetto dovuti al continuo fallimento del task.

Il calcolo della velocità istantanea avviene all'interno del ciclo *FixedUpdate*, garantendo la sincronizzazione con il motore fisico di Unity che opera a una frequenza costante di 50 Hz (corrispondente a un intervallo di 20 ms). Lo spostamento rilevato tra *frame* consecutivi viene rapportato al tempo trascorso tramite il metodo *Time.fixedDeltaTime* di Unity. Per attenuare eventuali artefatti o micro-oscillazioni nel segnale, la velocità viene processata attraverso una media mobile calcolata su un *buffer* FIFO di cinque campioni. Il sistema tiene traccia della velocità massima registrata, utilizzando tale valore per aggiornare la dimensione della barra, calcolata dalla funzione *custom SpeedToPixels.* Questo metodo normalizza l'input rispetto alla velocità target moltiplicata per un fattore di scala pari a 2 (denominato *maxMultiplier*) e restituisce

in uscita il valore normalizzato moltiplicato per la lunghezza del contenitore. Il fattore di scala determina la saturazione del feedback: esso definisce il valore massimo di velocità, ovvero il doppio della velocità target, per il quale la barra raggiunge il riempimento completo.

Sebbene il calcolo della velocità avvenga a 50Hz, l'interfaccia grafica viene aggiornata con una frequenza di 10 Hz per garantire una visualizzazione ottimale evitando traslazioni troppo rapide del biofeedback. Il movimento della barra è gestito dal sistema attraverso due metodi *custom* (*StartMovementTo* e *StepUI*) per garantire un'animazione fluida coerente con il *frame rate* dell'applicazione. La procedura ha inizio con il metodo *StartMovementTo*, che funge da pianificatore della transizione e viene invocato nel *FixedUpdate* ogni 10Hz se il sistema rileva un nuovo valore massimo di velocità. Questo metodo ha il compito di memorizzare la posizione di partenza dell'indicatore, aggiornare il nuovo valore di destinazione, ripristinare la variabile temporale (*Elapsed*) e definire la durata temporale entro cui il movimento deve completarsi, attivando contestualmente un *flag* interno che segnala al sistema l'avvio della progressione.

Successivamente, il calcolo effettivo dello spostamento avviene nell'*Update* attraverso il metodo *StepUI*. Utilizzando il tempo trascorso tra un *frame* e l'altro (metodo *built-in Time.deltaTime*), questo modulo calcola il rapporto tra il tempo trascorso dall'avvio della transizione (*Elapsed += Time.deltaTime*) e la durata impostata da *StartMovementTo*. Questo valore viene utilizzato per determinare la nuova posizione della barra effettuando un'interpolazione lineare (metodo *built-in Mathf.Lerp)* tra il punto di partenza e quello di arrivo della transizione. In seguito, il sistema provvede all'aggiornamento la posizione dell'indicatore agendo sulla componente x del parametro *sizeDelta*. Contestualmente, viene eseguita una verifica per determinare se l'estensione della barra rientri nella zona target: in caso positivo, la tonalità cromatica viene impostata su verde per fornire un rinforzo positivo, mentre uno scostamento dai limiti prefissati determina la commutazione istantanea verso il rosso, segnalando la condizione di errore. Trascorso il tempo prestabilito la barra viene forzata al punto di arrivo per correggere eventuali errori di interpolazione e vengono aggiornati gli indicatori interni che segnalano la fine della transizione.

La dimensione della barra viene riportata a zero ogni volta che si passa dalla fase di andata a quella di ritorno o se il biofeedback non viene aggiornato entro un certo intervallo di tempo. I cambiamenti di direzione sono individuati utilizzando la stessa logica implementata da *CircluarMovement* (*cfr.* sez. *"*Mapping da movimento lineare a movimento circolare*"*).

La riproduzione del segnale acustico in Unity è mediata dal componente nativo *AudioSource*, che funge da emettitore sonoro per una specifica risorsa audio (*AudioClip*). All'interno dell'architettura del progetto, il *FeedbackManager* coordina l'emissione sonora attraverso una *coroutine* dedicata, la quale implementa un ciclo di sospensione basato sul parametro t*imeAudiofeedbac*k. Utilizzando l'istruzione *yield return new WaitForSeconds*, la *coroutine* mette in pausa la propria esecuzione tra un'attivazione e l'altra, permettendo al metodo *Play*() *dell'AudioSource* di riprodurre il suono a intervalli temporali regolari. Questo approccio garantisce che il feedback uditivo (audio di un tamburo) mantenga una cadenza costante senza gravare sul ciclo di aggiornamento principale del software.

Biofeedback EMG

Il sistema di monitoraggio delle attivazioni muscolari, coordinato dalla classe *EMGFeedbckManager,* è finalizzato a identificare schemi motori scorretti, con particolare riferimento all'abduzione compensatoria della spalla durante l'esecuzione del task riabilitativo.

A differenza dei biofeedback cinematici precedentemente descritti, l'informazione visiva non viene trasmessa all'utente mediante indicatori grafici nel Canvas, bensì attraverso la manipolazione della posizione del remo. Qualora venga rilevata una condizione di compenso, ovvero un'attività elettromiografica anomala del muscolo target, l'elemento virtuale viene progressivamente orientato verso un assetto neutro. Come illustrato nella Figura 31 questa riconfigurazione geometrica fa sì che il remo virtuale si ritrovi fisicamente sollevato rispetto al pelo dell'acqua. Tale strategia fornisce un biofeedback visivo e, contestualmente, annulla l'azione propulsiva condizionando l'efficacia della remata all'assenza di strategie compensatorie. Infatti, anche se l'utente continua a svolgere il gesto riabilitativo, il remo rimarrà fuori dall'acqua finché non verrà ripristinato uno schema motorio corretto.

Figura 31: Sequenza di frame che mostrano l'effetto della coroutine per sollevare il remo dall'acqua quando vengono rilevati dei movimenti compensatori. In alto vista laterale della scena di test, in basso prospettiva dell'utente nell'exergame.

Per il corretto funzionamento all'interno dell'applicazione l' *EMGFeedbckManager* presenta un metodo di inizializzazione *custom* (*Initialize*). Durante questa procedura, il sistema riceve i riferimenti al componente *CircularMovements*, responsabile del tracciamento cinematico locale, e al *RowManager*, che gestisce la logica di spinta sincronizzata in rete. In base alla lateralità specificata, il metodo calcola dinamicamente l'offset da applicare al *CircularMovements*, inoltre esegue una serie di controlli di sicurezza, verificando se la sessione corrente prevede l'uso dell'EMG tramite *SessionInfo* e che la connessione alla *SyncStation* e i relativi *buffer* di dati siano disponibili. L'inizializzazione si conclude con il calcolo della soglia di attivazione del deltoide, determinata moltiplicando il valore della Massima Contrazione Volontaria (MVC), recuperato dal CalibrationData, per una percentuale pari al 20% [113].

A valle dell'inizializzazione viene abilitata la logica di valutazione dei compensi. Quest'ultima è eseguita ad ogni *frame* e si basa sul confronto tra l'inviluppo EMG, elaborato dal modulo *DuePlusDataContainer*, e la soglia di attivazione. Per evitare repentine oscillazioni dello stato del biofeedback e ridurre il numero di falsi positivi, il sistema adotta un meccanismo a doppia soglia, introducendo una soglia temporale oltre a quella sull'ampiezza del segnale [114]. L' *EMGFeedbckManager* esamina gli ultimi 5 campioni dell'inviluppo presenti all'interno del *buffer* di dati (corrispondente a una finestra temporale di 100 millisecondi a 50Hz). L'attivazione muscolare anomala viene confermata quando tutti i campioni della finestra risultano superiori alla soglia sull'ampiezza. In questo caso la classe chiama la coroutine *TransitionTo.* Tale procedura modifica dinamicamente i parametri del componente *CircularMovements,* per alterare la traiettoria del remo. Tale transizione riduce progressivamente il raggio della traiettoria circolare verso il valore minimo di 0.1 e trasla lateralmente l'offset spaziale, utilizzando le funzioni integrate *Vector3.Lerp* e *Mathf.Lerp* per garantire un movimento fluido tra lo stato iniziale e quello di errore. La progressione temporale è gestita attraverso l'accumulo del *Time.deltaTime*, che permette di calcolare un coefficiente di completamento normalizzato rispetto alla durata totale impostata a 0.5 s. Contestualmente all'avvio dell'animazione, viene aggiornato un *flag* di stato della classe *RowManager* associata al remo, azzerando istantaneamente ogni possibile contributo di forza alla dinamica della barca. Questo controllo di sicurezza si rivela fondamentale per prevenire comportamenti anomali nelle fasi iniziali o finali della transizione, ovvero negli istanti in cui la *coroutine* è già attiva ma il remo si trova ancora parzialmente immerso nel fluido. Lo stato di attivazione permane finché non vengono rilevati 5 campioni consecutivi sottosoglia. Una volta che il segnale sEMG rientra nei parametri operativi corretti, la medesima *coroutine* provvede a ripristinare il raggio e l'offset originario, riabilitando la logica di spinta e permettendo al soggetto di riprendere l'esercizio in modo efficace.Questa strategia di controllo permette di preservare la relazione tra il movimento lineare del paziente nel mondo reale e la rotazione del remo virtuale durante lo svolgimento dell'esercizio. Infatti, anche quando la remata viene resa inefficace, il *CircularMovements* continua ad aggiornare la posizione del *Game Object* di controllo sulla circonferenza, mentre il *RowManager* mantiene l'orientamento del remo verso quest'ultimo; tuttavia, poiché la circonferenza descritta risulta estremamente ridotta, il remo appare quasi immobile. Al ripristino delle condizioni ottimali, il ritorno del remo in posizione operativa avviene mantenendo la piena coerenza spaziale con il movimento eseguito dall'utente, garantendo una transizione fluida tra i due stati.

Modellazione e valutazione preliminare del BF basato su EMG

L'obiettivo di questa sezione è descrivere la creazione e validazione di un modello di regressione soggetto-specifico in grado di rilevare le sovra-attivazioni del muscolo deltoide laterale. Queste attivazioni sono riconducibili ad irrigidimenti del braccio o a movimenti compensatori di abduzione, ovvero strategie motorie alternative adottate dai soggetti patologici per completare il task di *reaching* a fronte di un deficit motorio o a causa di sinergie flessorie non fisiologiche [80], [81], [82], [84]. Essendo il modello specificatamente progettato per rilevare le variazioni di abduzione e non quelle relative all'adduzione, data la natura del compenso, è ragionevole applicare il modello di stima esclusivamente durante il tragitto verso il target.

Il fine di questo modello all'interno dell'applicazione di teleriabilitazione è quello di arricchire il biofeedback basato sul segnale elettromiografico per la rilevazione di tali compensi, passando da un approccio basato su soglia a uno soggetto-specifico e performance-specifico. Un secondo obiettivo è quello di fornire al soggetto non solo informazioni sul risultato (*Knowledge of Results)*, quindi informarlo quando sta attuando la strategia motoria scorretta, ma anche quantificarla, fornendogli un ritorno proporzionale alla severità del compenso *(Knowledge of Performance),* senza l'impiego di ulteriori sensori oltre a quello elettromiografico.

Partecipanti e apparato sperimentale

Per la raccolta dei dati finalizzata alla creazione del modello sperimentale sono stati reclutati 15 soggetti sani (9 maschi e 6 femmine) di età 24.7 ± 2.5 tutti con braccio dominante destro. L'apparato sperimentale consisteva sistema WaveX di Cometa S.r.l (Figura 32 A), dotato di sensori miniX indossabili provvisti di unità di misura inerziale (IMU) e sensori elettromiografici (EMG). La configurazione dei dispositivi, illustrata in Figura 32 B, prevedeva il posizionamento di una prima unità sul braccio dominante del soggetto (in modalità combinata IMU+EMG) e di una seconda a livello dello sterno (in modalità solo IMU), entrambe fissate sulla cute del soggetto tramite supporto adesivo. Due elettrodi Ag/AgCl con gel conduttivo sono stati applicati sul muscolo deltoide laterale, circa 3 cm sotto l'acromion e a una distanza reciproca di 2 cm, collegati al primo dispositivo. Durante lo svolgimento del protocollo sperimentale sono stati utilizzati due laptop: uno per la raccolta, visualizzazione e salvataggio dei dati EMG e angoli articolari utilizzando il software proprietario di Cometa (EMGandMotionTools); l'altro, collegato a un televisore posto frontalmente al partecipante, per la visualizzazione dell'esercizio motorio guidato da un avatar animato, sviluppato in Unity, per guidare la temporizzazione del task motorio (Figura 32 C). Gli angoli articolari sono stati calcolati in maniera automatica dal software impostando il Sensor Data Protocol a EMG 2 kHz Fused6xData 100 Hz per il sensore EMG+IMU e Fused6xData 400 Hz per il sensore posto sullo sterno (solo IMU). Quest'ultimi comprendevano l'angolo descritto dal segmento tronco-braccio nel piano coronale (abdo-adduzione), sagittale (flesso-estensione verticale) e trasversale (flesso-estensione orizzontale). La frequenza di campionamento dei segnali EMG è 2 kHz mentre quella degli angoli articolari è 100Hz, tuttavia, il software durante l'esportazione forniva entrambi i dati a 2kHz, replicando il valore degli angoli (20 campioni per ogni angolo). Prima dell'inizio del test i soggetti sono stati istruiti sull'esecuzione degli esercizi e hanno firmato il consenso informato per il trattamento dei dati personali

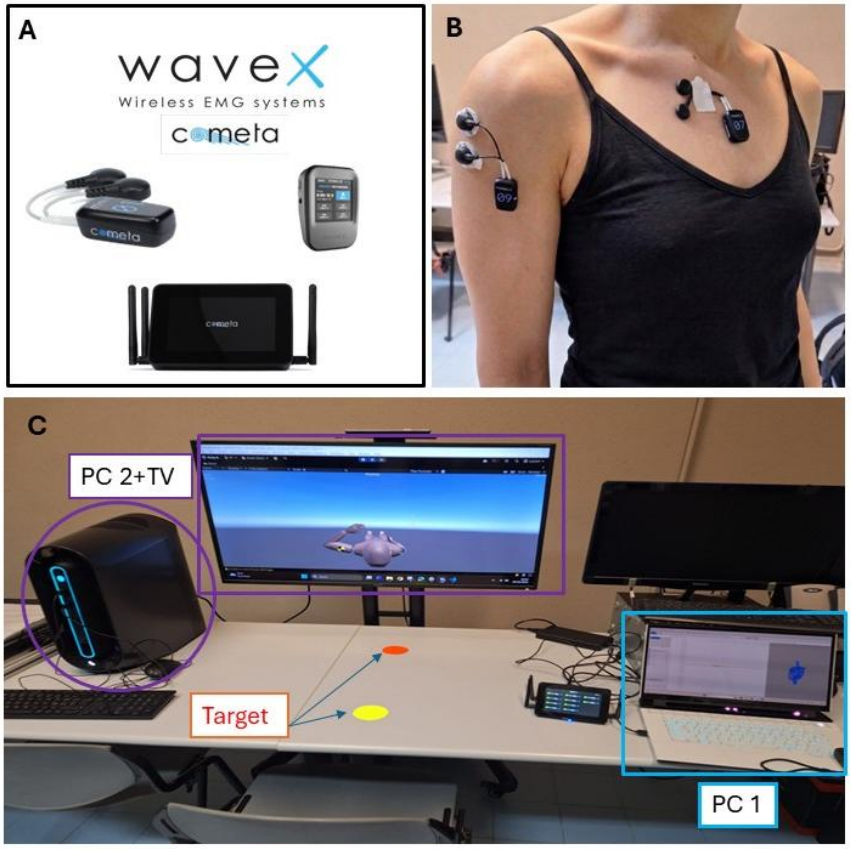

Figura 32: Apparato e set-up sperimentale. A) Sistema waveX di Cometa S.r.l: a sinistra i sensori miniX dotati di EMG e IMU, a destra il RemoteX per gestire a distanza l'acquisizione, al centro il sistema di ricezione e comunicazione con le sonde. B) Posizionamento dei due sensori miniX: il primo sul braccio dominante in modalità IMU+EMG con elettrodi posti lungo il capo laterale del deltoide (3cm sotto l'acromion, IED 2 cm) il secondo nello sterno in modalità solo IMU. C) Set-up per l'esecuzione degli esercizi: PC 1 per la raccolta dei dati, PC2 + TV per la visualizzazione e temporizzazione del task e i target per l'esecuzione del reaching in posizione seduta.

Protocollo sperimentale

Il protocollo sperimentale prevedeva una fase preliminare di calibrazione dei sensori, la stima della contrazione massimale volontaria (*MVC*) e 4 esercizi: il primo esercizio consisteva in movimenti di abdo-adduzione di spalla, eseguiti stando in piedi, muovendo il braccio destro sul piano sagittale, ed era finalizzato alla creazione (training) del modello; gli esercizi successivi consistevano in movimenti di *reaching* planare (*point-to-point*), svolti da seduti di fronte ad un tavolo, ed erano finalizzati al test del modello.

La calibrazione dei sensori IMU è stata effettuata mediante l'utilizzo del software proprietario, che prevedeva il posizionamento dei dispositivi in una configurazione di riferimento su una superficie piana per la correzione di eventuali drift del sistema. Successivamente, una volta posizionati i sensori sul soggetto, a quest'ultimo veniva richiesto di posizionarsi frontalmente al tavolo e assumere la T-pose (abduzione di spalla sul piano frontale di 90 gradi con entrambe le braccia, Figura 33 A), al fine di consentire una corretta ricostruzione della cinematica articolare da parte del software.

Successivamente è stata acquisita l'MVC del muscolo deltoide laterale utilizzando un protocollo simile a quello proposto da Al-Qaisi et al. (2015) [115]. La contrazione massimale è stata ricavata utilizzando delle bande elastiche con resistenza variabile, scelte in base alle capacità muscolari del paziente, ancorate alla parete ad un'altezza di circa 60cm. Il soggetto posizionato lateralmente rispetto alla parete, impugna la banda elastica con la mano dominante ed esegue un'abduzione con la massima forza possibile fino a circa 90 gradi con il gomito completamente esteso per massimizzare l'attivazione [116] (Figura 33 B). La contrazione isometrica viene mantenuta per un totale di 8 s considerando 3-5 s per raggiungere l'MVC e 3 s di mantenimento [115]. Questa procedura è stata ripetuta 3 volte, con un riposo di 2 minuti tra una ripetizione e l'altra per evitare fenomeni di fatica.

Il primo esercizio, finalizzato alla creazione del modello, consisteva in 3 movimenti di abduzione-adduzione dell'arto superiore destro sul piano frontale e 3 movimenti di flesso-estensione sul piano sagittale eseguiti seguendo i movimenti dell'avatar (Figura 33 C). La finalità dell'avatar era quella di guidare il soggetto nell'esecuzione di movimenti lenti e fluidi con una cinematica nota; in particolare, il soggetto partiva da una posizione neutra (N-pose, con le braccia estese lungo il corpo), fino alla completa estensione dell'arto (~150°) e ritornava alla posizione iniziale, con una durata totale di 4 s (2 s di abduzione/flessione e 2 s di adduzione/estensione). Questa fase aveva un duplice scopo: il primo era verificare il corretto posizionamento dei sensori ed eventuali fenomeni di cross-talk in quanto il capo laterale del deltoide è il principale attuatore dell'abduzione, mentre quello anteriore interviene maggiormente nella flessione; il secondo era acquisire dalle ripetizioni di abdo-adduzione i dati necessari per la creazione del modello.

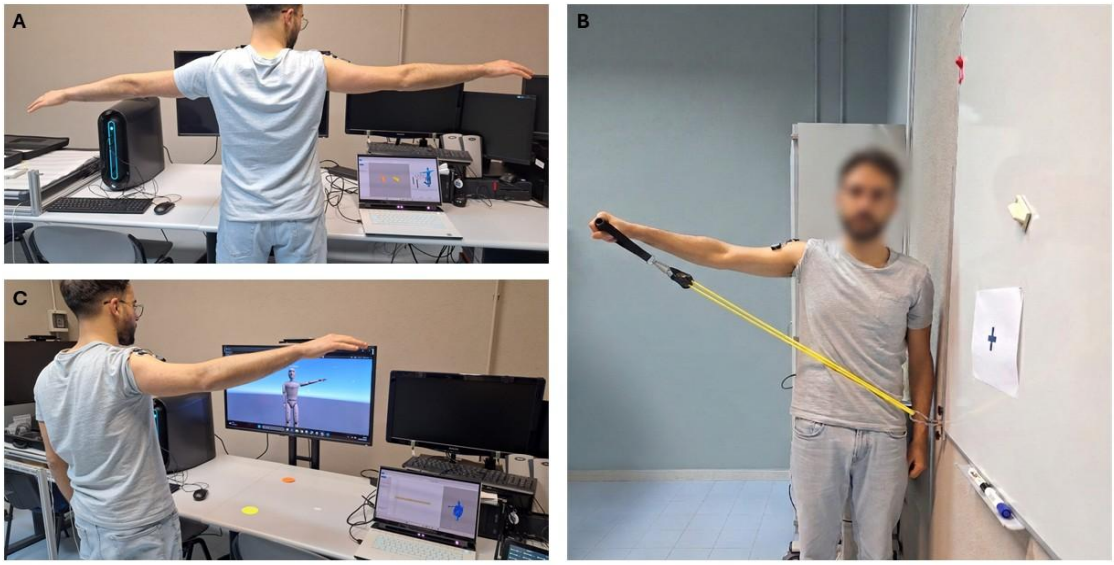

Figura 33: A) T-pose (braccia abdotte di 90°) assunta dal soggetto per la calibrazione del software di ricostruzione articolare (EMGandMotionTools, Cometa S.r.l). B) Acquisizione della massima contrazione volontaria (MVC) tramite abduzione isometrica a ~90° eseguita stirando una banda elastica ad elevata resistenza. C) Esercizio di abdo-adduzione pura compiuto seguendo i movimenti dell'avatar.

Per testare il modello sono stati progettati 3 esercizi di *reaching* svolti in condizioni differenti:

1. **Task 1**: il movimento è eseguito con il gomito appoggiato sul tavolo (Figura 34 A), in questa condizione il deltoide risulta a riposo in quanto il peso dell'arto è scaricato sul piano.
2. **Task 2:** il movimento è eseguito con il gomito sollevato di qualche centimetro rispetto al tavolo (Figura 34 B), in queste condizioni il deltoide è parzialmente attivato per stabilizzare il braccio contro gravità.
3. **Task 3:** il movimento è eseguito simulando movimenti compensatori: durante l'esercizio del *reaching*, al soggetto è richiesto di mantenere una piccola bottiglia vuota e posizionarla sul target finale, ruotandola di 180 gradi durante il movimento (Figura 34 C). Questa richiesta costringe il soggetto a sollevare e ruotare il gomito verso l'asse sagittale in modo prevedibile e ripetibile, simulando il movimento compensatorio di abduzione di spalla ed intra-rotazione del gomito, tipico dei soggetti patologici.

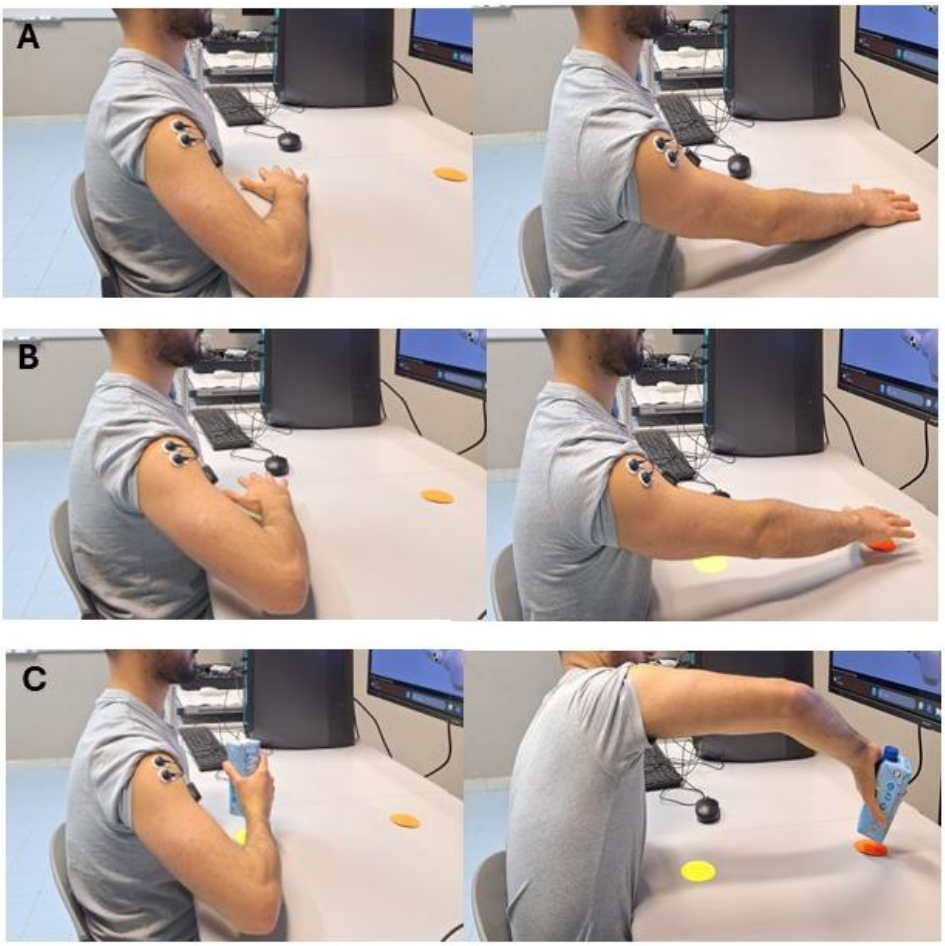

Figura 34: I tre esercizi di reaching planare (point-to-point) per la validazione del modello. (A) Reaching in appoggio (task 1): durante l'esecuzione il peso del braccio viene supportato dall'appoggio dell'avambraccio sul tavolo, il deltoide laterale risulta quindi a riposo. B) Reaching senza appoggio (task 2): durante l'esecuzione il braccio risulta leggermente sollevato, il deltoide laterale sostiene il peso del braccio (parzialmente attivo). C) Reaching con movimenti compensatori (task 3): durante l'esecuzione del reaching viene chiesto al soggetto di ruotare la bottiglia per indurre l'abduzione e l'intra-rotazione del gomito tipiche dei movimenti compensatori dei soggetti patologici.

Il protocollo prevedeva una prima fase di familiarizzazione (10 trials) e successivamente l'esecuzione vera e propria degli esercizi (30 trials per ognuno). Tra un esercizio e l'altro in entrambe le fasi seguiva una fase di riposo di 1 minuto. Durante l'esecuzione del task, l'avatar, mostrato sul monitor, eseguiva dei movimenti di *reaching* con una tempistica predefinita (1,5 s in andata e 1,5 s in ritorno).

I dati delle varie prove (MVC, *training* modello, familiarizzazione ed esecuzione del task) sono stati esportati in file di testo ed elaborati in MATLAB R2025b. I file comprendevano gli istanti temporali, i dati EMG del deltoide laterale, i quaternioni dei due IMU e gli angoli articolari calcolati automaticamente dal software (abduzione-adduzione, flessione-estensione orizzontale e verticale).

Valutazione preliminare dell'attività muscolare durante l'abdo-adduzione

Preliminarmente alla definizione del modello, è stata condotta un'analisi per caratterizzare il comportamento del deltoide, in termini di attivazione muscolare in funzione dell'escursione angolare, durante l'esecuzione delle 3 ripetizioni di abdo-adduzione di spalla. Il processo di elaborazione ha previsto inizialmente l'estrazione dell'inviluppo dai segnali EMG grezzi, al fine di determinare il profilo di attivazione muscolare. Successivamente, gli inviluppi ottenuti sono stati normalizzati rispetto alla Massima Contrazione Volontaria (%MVC), passaggio necessario per standardizzare i livelli di attivazione tra soggetti diversi. Le tre prove sperimentali, così elaborate, sono state segmentate distinguendo la fase di abduzione da quella di adduzione. Infine, è stata calcolata la media di insieme tra tutti i soggetti per ottenere la curva caratteristica del segnale in funzione dell'angolo articolare.

Inviluppo del segnale EMG

Il segnale EMG grezzo è stato sottoposto a una catena di elaborazione per la rimozione di eventuali artefatti e l'estrazione dell'inviluppo. La procedura si articola in tre fasi principali: filtraggio passa-banda, calcolo del valore efficace (RMS) e sottocampionamento con media mobile:

1. **Filtraggio passa-banda**: finalizzato a rimuovere le componenti di rumore e gli artefatti di movimento, preservando il contenuto informativo utile dell'attivazione muscolare. Per l'elettromiografia di superficie, il filtraggio tipico è 10-20 Hz e 450-500 Hz [117], [118]. A tale scopo, sono stati applicati in cascata due filtri a risposta impulsiva finita (FIR) di tipo *Equiripple*, con fase lineare [118], il primo passa-basso (ordine 127, frequenza di taglio a 440 Hz, frequenza di stop a 480 Hz e attenuazione di 80 dB), mentre il secondo passa-alto (ordine 150, frequenza di stop a 5 Hz, frequenza di taglio a 25 Hz e attenuazione 40dB).
2. ***Root Mean Square* (RMS)**: il segnale filtrato è stato trasformato nel suo inviluppo tramite il calcolo del *Root Mean Square* (RMS) su finestra scorrevole (*Sliding Window*) [119]. L'algoritmo calcola il valore RMS per ogni campione *i-esimo* secondo la formula:

$$RMS[i] = \sqrt{\frac{1}{W}\sum_{k=i}^{i+W-1} xp[k]^2} \tag{5}$$

   dove W è la lunghezza della finestra temporale, impostata a 640 campioni (pari a 320 ms) e *xp* è il segnale filtrato sul quale è stata eseguita un'operazione di *padding*, replicando il primo e l'ultimo campione per una lunghezza pari a metà finestra. Infine, il segnale risultante è stato riallineato temporalmente mediante uno scorrimento circolare (*circshift*) negativo di W/2 campioni
3. **Smoothing:** L'ultimo stadio della catena di elaborazione ha l'obiettivo di ridurre la variabilità del segnale RMS, tramite l'applicazione di un filtro di Savitzky–Golay [120], con ordine 4 e finestra di 621 campioni (≈310 ms), compatibile con le velocità e le frequenze in gioco dei segnali acquisiti.

Successivamente, l'inviluppo EMG è stato normalizzato dividendo i valori per la rispettiva MVC, esprimendo così l'attivazione muscolare come percentuale della massima capacità volontaria del soggetto. Il valore della MVC è stato determinato selezionando il picco massimo dell'inviluppo EMG tra le tre prove eseguite da ciascun soggetto.

Segmentazione delle fasi di abduzione-adduzione

Per la segmentazione delle fasi di abduzione e adduzione è stata eseguita preliminarmente una selezione manuale degli intervalli di interesse tramite ispezione visiva del tracciato temporale. Tale operazione si è resa necessaria per discriminare e separare le serie relative al movimento di abduzione-adduzione da quelle di flesso-estensione, registrate sequenzialmente all'interno della stessa acquisizione. La porzione di segnale angolare isolata è stata successivamente riscalata in un intervallo positivo al fine di correggere l'offset introdotto dal software di ricostruzione articolare (EMGandMotionTools), ridefinendo lo zero in corrispondenza della posizione iniziale N-pose (braccio lungo il corpo) anziché nella configurazione T-pose (braccio a 90°). L'algoritmo di segmentazione si basa sull'analisi della velocità angolare per individuare gli istanti di inversione del moto.

Sebbene la frequenza nativa del sensore IMU fosse di 100 Hz, l'esportazione a 2 kHz generava una ridondanza di dati caratterizzata da un andamento a gradino, compromettendo l'analisi della derivata. Per eliminare questo effetto e al contempo migliorare la qualità del segnale, è stato applicato un filtro di Savitzky–Golay, di ordine 4 e finestra di 621 campioni (≈310 ms) al segnale a 2 kHz. Al fine di minimizzare gli effetti di bordo, prima dell'applicazione del filtro è stato eseguito un *padding* del segnale replicando il primo e l'ultimo valore per una lunghezza pari al 10% del segnale. I campioni aggiunti sono stati opportunamente rimossi dopo il filtraggio. Successivamente, il filtro è stato applicato nuovamente con gli stessi parametri, ma configurato per calcolare la derivata prima del segnale.

L'identificazione dei limiti di inizio e fine di ogni fase (abduzione e adduzione) avviene mediante la rilevazione dei passaggi per lo zero (zero-crossing, ZC) della derivata. Per garantire la robustezza dell'algoritmo contro false rilevazioni dovute a esitazioni motorie o rumore, è stato imposto un vincolo di distanza temporale minima tra due attraversamenti consecutivi di 1,3 secondi, considerando le tempistiche dei movimenti imposte dal protocollo sperimentale. Qualora vengano identificati zero-crossing multipli all'interno di tale finestra, l'istante temporale viene determinato calcolando la media tra le occorrenze ravvicinate, risolvendo così eventuali ambiguità locali. Per ogni ripetizione vengono individuati 3 istanti corrispondenti all'inizio del movimento, al passaggio dalla fase di abduzione a quella di adduzione e alla fine del movimento.

Le 45 ripetizioni di abduzione e adduzione (15 soggetti per 3 ripetizioni) sono state ricampionate alla dimensione della ripetizione più lunga trovata dall'algoritmo di segmentazione. Successivamente è stata calcolata una ripetizione media effettuando la media delle ripetizioni intra-soggetto e successivamente la mediana delle ripetizioni medie dei soggetti (media inter-soggetto), per valutarne l'andamento a livello di popolazione.

Evidenze rilevate dall'analisi

Dall'analisi dell'andamento dell'inviluppo EMG normalizzato in funzione delle variazioni dell'angolo di abdo-adduzione (Figura 35) emergono alcune evidenze fondamentali per la costruzione del modello. Innanzitutto, l'andamento delle due fasi del movimento, ovvero quella eccentrica (abduzione) e quella concentrica (adduzione), è nettamente differente. Dal grafico emerge come il segnale elettromiografico non ripercorre lo stesso percorso durante il ritorno alla posizione iniziale, manifestando un fenomeno di isteresi [121]. Inoltre, sono presenti degli andamenti severamente non monotoni (ripiegamento del segnale su sé stesso) soprattutto nella parte di inversione del moto, che renderebbero impossibile trovare una relazione biunivoca tra attivazione muscolare e angolo. Da questa analisi emergono una serie di ipotesi cardine per lo sviluppo del modello:

1. Sono necessarie due funzioni di fitting per le diverse fasi del moto (abduzione e adduzione). Nell'applicazione reale del modello siamo interessati a valutare l'abduzione durante la fase di estensione (andata) e non di ritorno in posizione iniziale. Pertanto, nelle analisi successive verrà considerata solo la parte relativa all'abduzione.
2. Per garantire una ricostruzione fedele dell'andamento dell'attivazione muscolare in funzione dell'angolo attraverso una funzione di fitting è fondamentale limitare le componenti non monotone che si manifestano specialmente nella parte iniziale e finale del movimento.

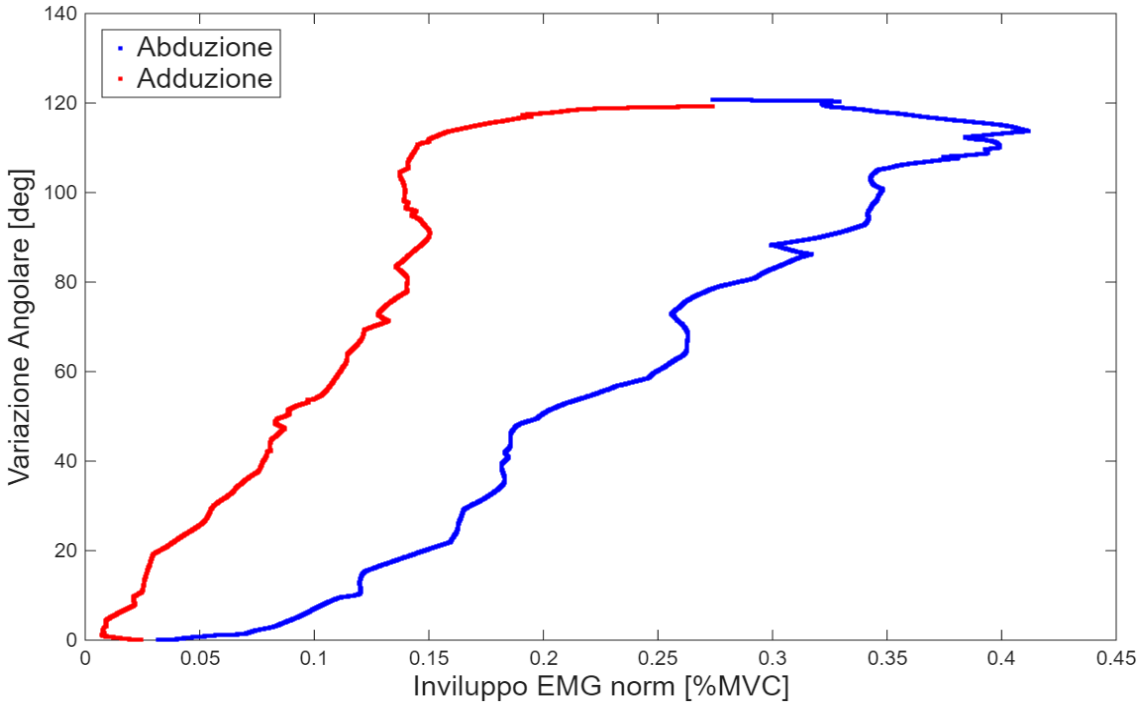

Figura 35: Andamento della relazione media dell'inviluppo EMG normalizzato in funzione delle variazioni dell'angolo di abdo-adduzione calcolata su tutto il dataset sperimentale. In blu il segnale relativo alla fase di abduzione, in rosso quello relativo alla fase di adduzione.

Creazione del modello

L'obiettivo del modello proposto è definire una funzione di fitting in grado di stimare le variazioni angolari a partire dall'inviluppo dell'attivazione muscolare del deltoide. Le ripetizioni (3 per ciascun soggetto), elaborate e segmentate come descritto nel paragrafo precedente, sono successivamente sottoposte ad un'analisi volta a escludere dal segnale le componenti non monotone, che comprometterebbero la qualità del fitting. Il criterio di esclusione si basa sull'ipotesi che le componenti non monotone significative si manifestano nelle fasi transitorie di accelerazione e decelerazione del movimento come diretta conseguenza del controllo di stabilizzazione motoria in queste due fasi. In funzione di questo vengono in primo luogo escluse dall'analisi le ripetizioni di abduzione che presentano un numero elevato di picchi nel profilo di velocità, con conseguente presenza di tratti eccessivamente non monotoni nella parte centrale del segnale (ripetizioni ritenute "non valide").

In particolare, un movimento di abduzione fluido presenta un solo picco di velocità angolare. La procedura di validazione accetta al massimo due picchi nel profilo di velocità, dovuti ad una eventuale decelerazione intermedia in T-pose e successiva riaccelerazione, che alcuni soggetti manifestano. Tali fasi sono individuate analizzando il profilo di velocità e accelerazione angolare, calcolate rispettivamente come derivata prima e seconda del segnale di posizione angolare, calcolate con il medesimo filtro utilizzato per la segmentazione (*padding* e filtro Savitzky–Golay, ordine di filtraggio 4 e finestra di 621 campioni, e ordine di derivazione 1). I picchi di velocità vengono individuati tramite analisi dello zero crossing discendenti (transizione da valore positivo a valore negativo) dell'accelerazione, escludendo eventuali falsi positivi dovuti all'avvio del movimento, corrispondente ai primi 0,15 s (300 campioni). Successivamente, se sono presenti più picchi, viene selezionato il secondo in ordine temporale e individuata la valle successiva tramite zero-crossing ascendente (transizione da valore negativo a positivo). La ripetizione è considerata valida solo se la valle individuata corrisponde alla fine del movimento, ovvero se presenta un'ampiezza minore al 15% della velocità di picco o corrisponde alla fine del segnale [122].

Sulle sole ripetizioni ritenute valide è stata successivamente applicata una segmentazione al fine di isolare la parte centrale del movimento ed eliminare le code non monotone. Per quanto riguarda la determinazione di questi punti, è stato adottato un approccio ibrido che integra informazioni cinematiche ed elettromiografiche. Nello specifico, il criterio cinematico identifica il momento in cui la velocità angolare supera il 10% del picco massimo per il punto inziale, analogamente il punto finale corrisponde all'istante in cui il segnale decresce al di sotto della medesima soglia [122]. L'analisi sull'attivazione muscolare invece, include l'istante corrispondente al minimo assoluto dell'inviluppo EMG, per l'inizio del movimento, mentre per il punto finale viene selezionato il primo picco EMG che supera una soglia di ampiezza relativa al 90% del massimo locale. Quest'ultima condizione è giustificata dal fatto che, durante il movimento di abduzione, l'attivazione del deltoide mediale aumenta all'aumentare dell'angolo di abduzione ed eventuali ripiegamenti del segnale sono dovuti alle attivazioni di controllo muscolare che precedono la fine o l'inizio del movimento e servono a stabilizzarlo [123]. L'algoritmo seleziona la condizione più conservativa dei due criteri, ovvero l'evento che si verifica più tardi per la parte iniziale e quello che si verifica prima per quella finale.

Il passo successivo ha riguardato la selezione della funzione più appropriata nel rappresentare le variazioni dell'attivazione muscolare. Attraverso il *Curve Fitting Toolbox* di MATLAB, sono state valutate varie tipologie di *fitting* per modellare l'inviluppo EMG in funzione della variazione di angolo di abduzione. Le funzioni di *fitting* analizzate sono riportate in Tabella 1.

Tabella 1: Funzioni di fitting analizzate per la creazione del modello

Tipologia di fitting	***Formula***	***N° parametri***	***Punti iniziali***
Polinomiale	$f(x) = p_1 x + p_2$	2	*Automatico* (Linear Least Squares)
Polinomiale2	$f(x) = p_1 x^2 + p_2 x + p_3$	3	*Automatico (Linear Least Squares)*
Polinomiale3	$f(x) = p_1 x^3 + \cdots + p_4$	4	*Automatico* (Linear Least Squares)
Polinomiale4	$f(x) = p_1 x^4 + \cdots + p_5$	5	*Automatico* (Linear Least Squares)
Esponenziale	$f(x) = ae^{bx} + ce^{dx}$	4	[38.4940, 1.8500, 0, 1.8500]
Logistica	$f(x) = \frac{a}{1 + e^{-k(x-x_0)}}$	3	[105.6741, 19.0422, 0.2707]
Logistica2	$f(x) = d + \frac{a-d}{1 + (x/c)^b}$	4	*self-starting* [124]
Fourier	$f(x) = a_0 + a_1 cos(wx) + b_1 sin(wx)$	4	[0, 0, 0, 0]
Fourier2	$f(x) = a_0 + a_1 cos(wx) + b_1 sin(wx) + a_2 cos(2wx) + b_2 sin(2wx)$	6	[0, 0, 0, 0, 0, 0]

Dal punto di vista computazionale, la modellazione ha richiesto strategie differenziate in base alla natura matematica delle funzioni. Mentre i modelli polinomiali beneficiano di un'inizializzazione automatica basata sui minimi quadrati lineari (*Linear Least Squares*) e non necessitano pertanto di punti di partenza espliciti, per le altre funzioni è stato necessario definire dei valori iniziali specifici per garantire la convergenza dell'algoritmo. Per la maggior parte di queste ultime, i punti di partenza sono stati determinati empiricamente analizzando i parametri stimati internamente dal toolbox sull'intero gruppo di soggetti. Tuttavia, tale approccio generalista non si è rivelato efficace per la funzione Logistica2 a 4 parametri, per la quale l'elevata variabilità inter-soggetto ha impedito l'individuazione di una configurazione iniziale univoca valida per tutti i partecipanti.

Per superare questa limitazione, è stata implementata una routine di inizializzazione specifica per ogni soggetto. I valori iniziali dei parametri (asintoti, pendenza e punto di flesso) sono stati stimati direttamente dalle proprietà geometriche dei dati sperimentali, in accordo con l'interpretazione fisiologica del modello descritta da Motulsky & Christopoulos (2004) [124].

La valutazione del modello ottimale è stata condotta basandosi sull'analisi della qualità della regressione (*Goodness of Fit*) e della complessità del modello (numero di parametri). La bontà del fitting è stata valutata utilizzando il Root Mean Squared Error (RMSE) e una versione modificata dell'R^2, ovvero l'adjusted R^2 [125]. L'R^2 misura il grado di successo del fitting nel rappresentare la variazione dei dati, ovvero il quadrato della correlazione tra i valori reali e i valori stimati dal modello. Questa metrica può assumere qualsiasi valore tra 0 e 1, con valori più vicini all'unità che indicano un fitting migliore, ed è definita come il rapporto tra la somma dei quadrati della regressione (SSR) e la somma totale dei quadrati (SST):

$$SSR = \sum_{i=1}^{n} w_i(\hat{y_i} - \bar{y})^2 \tag{6}$$

$$SST = \sum_{i=1}^{n} w_i(y_i - \bar{y})^2 \tag{7}$$

dove y rappresenta i dati reali, $\bar{y}$ la loro media e $\hat{y}$ i valori predetti dal modello.

Introducendo la definizione di Somma dei quadrati degli errori (SSE), nota anche come somma dei quadrati dei residui:

$$SSE = \sum_{i=1}^{n} w_i(y_i - \hat{y_i})^2 = SST - SSR \tag{8}$$

si ottiene la seguente formulazione dell' R^2:

$$R^2 = \frac{SSR}{SST} = 1 - \frac{SSE}{SST} \tag{9}$$

La scelta di utilizzare la versione modificata deriva dal fatto che questa metrica tende ad aumentare al crescere del numero di coefficienti del modello, anche quando il miglioramento del fitting non è significativo [125]. Questa tendenza potrebbe portare a degli errori nel confronto tra modelli con un numero differente di parametri (se il numero di parametri cresce, il modello sicuramente performa meglio, ma è anche più complesso). Per ovviare a questo problema è stato valutato l'$Adjusted\ R^2$.

A differenza della versione standard, questa metrica penalizza l'aggiunta di parametri superflui tenendo in considerazione i gradi di libertà residui v , definiti come $v = n - m$, dove n è il numero di osservazioni e m il numero di coefficienti stimati dal modello. La formula dell' $Adjusted\ R^2$ è dunque espressa come:

$$Adjusted\ R^2 = 1 - \frac{SSE(n-1)}{SST(v)} \tag{10}$$

L'RMSE, noto anche come errore standard della regressione, stima la deviazione standard della componente casuale nei dati, ovvero l'errore quadratico medio tra i valori osservati e quelli predetti dal modello ed è definito come:

$$RMSE = \sqrt{\frac{SSE}{v}} \tag{11}$$

Applicazione del modello

Ogni ripetizione valida è stata trattata come un modello cinematico indipendente, la cui funzione ottenuta dal fitting è stata successivamente applicata ai dati elettromiografici registrati durante i tre task di *reaching* svolti dal soggetto (esercizi di test). Per i motivi spiegati in precedenza, il modello è stato applicato solo durante i movimenti di estensione; pertanto, è stato necessario implementare un algoritmo dedicato alla separazione della fase di andata (verso il target) da quella di ritorno (verso lo *starting point*).

La procedura di segmentazione si basa sull'analisi dell'angolo di flessione orizzontale della spalla e della sua derivata temporale (velocità angolare), sfruttando i vincoli temporali imposti dal protocollo sperimentale e la quasi-periodicità del movimento di *reaching* sul piano trasversale, in cui il movimento di flesso-estensione si esplica. Il segnale angolare è stato inizialmente sottoposto ad un *unwrapping*, tramite la funzione *built-in unwrap* di Matlab, per correggere eventuali discontinuità dovute alla rappresentazione degli angoli nell'intervallo [-π, π]. La funzione rileva le differenze tra angoli successivi e, se maggiori di una soglia predefinita (di *default* π), aggiunge o sottrae multipli di 2π finché la discontinuità non viene eliminata. Successivamente, sono state rimosse dai segnali angolari potenziali derive (*drift)* lineari, non correttamente compensate dal software utilizzato per la ricostruzione cinematica, dovute prevalentemente all'integrazione del *bias* (errore) costante del giroscopio dei sensori inerziali [126]. La deriva residua è stata rimossa applicando un *de-trending* del segnale tramite la funzione *built-in detrend* di MATLAB, quest'ultima stima la retta di regressione lineare che meglio approssima l'andamento del segnale attraverso il metodo dei minimi quadrati e la sottrae puntualmente ai dati in ingresso. La velocità angolare, utilizzata per identificare le fasi moto, è stata calcolata tramite derivata del segnale angolare pre-processato, attraverso un filtro di Savitzky-Golay (ordine 4, finestra di 621 campioni). L'algoritmo identifica gli istanti temporali relativi ai picchi di velocità, i quali segnano la transizione tra la fase di accelerazione e quella di decelerazione durante il raggiungimento del target. Tali eventi temporali sono identificati analizzando gli attraversamenti dello zero (*Zero-Crossing*, ZC) discendenti (transizione da valore positivo a valore negativo) dell'accelerazione, calcolata a sua volta applicando il medesimo filtro derivativo alla velocità. Tutti picchi inferiori a una soglia adattiva pari al 40% del massimo del segnale vengono rimossi, in quanto riconducibili a movimenti non fluidi o artefatti piuttosto che a effettive inversioni del moto. In seguito, l'algoritmo adotta una strategia di *clustering* per la gestione dei picchi multipli: in accordo con la cinematica del movimento (durata teorica della ripetizione pari a 3s), vengono accorpati i picchi con una distanza inferiore a 2 s, interpretando tali oscillazioni come parte di una singola ripetizione anziché come eventi distinti. All'interno del

cluster, vengono identificati il primo e l'ultimo picco, che fungono da perni per la delimitazione delle fasi di andata e di ritorno (in caso di cluster con un solo elemento questi punti coincidono). L'inizio dell'andata viene individuato nell'istante corrispondente alla prima valle di velocità (ZC ascendente dell'accelerazione) antecedente al primo evento del cluster in una finestra di ricerca compresa tra 0.5 e 2.2 secondi. Se vengono rilevate più valli l'algoritmo rimuove quelle troppo distanti temporalmente (> 0.8 s) rispetto alla più vicina al picco. Simmetricamente, la fine della fase di ritorno è determinata ricercando l'ultima valle dopo l'ultimo picco del cluster in una finestra di ricerca analoga alla precedente.

Successivamente alla segmentazione temporale, in maniera analoga a quanto fatto per la creazione del modello, è stato applicato un ulteriore criterio di selezione cinematica per ogni fase di andata. In particolare, sono stati esclusi dall'analisi gli istanti iniziali e finali del movimento in cui la velocità angolare istantanea risultava inferiore al 10% della velocità di picco registrata nella specifica ripetizione. La velocità è stata calcolata mantenendo i medesimi parametri di filtraggio utilizzati per la segmentazione.

I valori dell'inviluppo EMG normalizzato relativi alle fasi di andata sono stati sottoposti a un processo di saturazione, ovvero limitati all'interno dell'intervallo dei valori utilizzati per la creazione del modello. Tale operazione è fondamentale per prevenire comportamenti anomali, qualora fossero registrati valori di attivazione muscolare al di fuori dell'intervallo di *fitting*. L'angolo di abduzione è stato quindi stimato applicando il modello matematico ai dati EMG pre-elaborati mediante la funzione *feval* del *Curve Fitting Toolbox* di MATLAB. Tale funzione valuta l'oggetto *cfit* (che incapsula l'equazione del modello e i coefficienti ottimali) in corrispondenza del vettore in ingresso (l'EMG), restituendo la risposta predetta del sistema.

Per ciascuna ripetizione, è stato sottratto alla serie temporale il valore del primo campione, in modo che i valori angolari fossero rappresentati in termini di variazioni angolari. La medesima operazione è stata applicata all'angolo reale misurato dai sensori inerziali. Questa procedura è stata adottata poiché l'obiettivo dell'analisi non è la valutazione della posizione assoluta dell'arto, bensì la quantificazione escursione angolare relativa eseguita durante il task, eliminando così l'influenza di eventuali offset o differenze posturali iniziali.

Successivamente, nell'ambito di ciascuna fase di andata, è stata operata una distinzione puntuale tra gli intervalli di abduzione e quelli di adduzione. Tale discriminazione si rende necessaria in quanto il modello è stato strutturato per stimare esclusivamente le variazioni angolari in abduzione, direttamente correlate alle attivazioni anomale del deltoide laterale. Durante i movimenti compensatori patologici, così come in quelli simulati (task 3), la presenza di transitori di adduzione è verosimilmente imputabile a movimenti di *reaching* non eseguiti correttamente (per esempio con il gomito allineato con la traiettoria di movimento, pertanto addotto continuamente in flessione), artefatti nella ricostruzione cinematica dei piani di movimento dovuto ai limiti intrinseci dei sensori IMU, o a imprecisioni nella segmentazione del *reaching* sulla base dell'angolo di flesso-estensione del gomito. Pertanto, l'isolamento delle sole fasi monotone crescenti consente di valutare l'effettiva capacità predittiva del modello sulla dinamica di interesse, al netto di incoerenze cinematiche esecutive e/o strumentali.

Compressione della dinamica: Time Warping

Durante l'esecuzione del gesto riabilitativo, i movimenti compensatori di abduzione tendono a manifestarsi con dinamiche più rapide rispetto ai movimenti controllati utilizzati per l'addestramento del modello. Questa discrepanza di velocità introduce una criticità nella stima cinematica. Come supportato dallo studio di Alpert et al. (2000) [127], la velocità di esecuzione influenza significativamente il pattern elettromiografico: movimenti rapidi comportano, rispetto a quelli lenti, una variazione della dinamica del segnale e un'anticipazione del picco di attivazione (Figura 36). Ne consegue che, specialmente nella fase iniziale del movimento (primi 30-60°), a parità di escursione angolare si registrano livelli di attivazione muscolare superiori per generare la necessaria accelerazione. Poiché il modello è stato calibrato su movimenti più lenti, tale incremento di attivazione viene interpretato erroneamente come una variazione angolare più ampia.

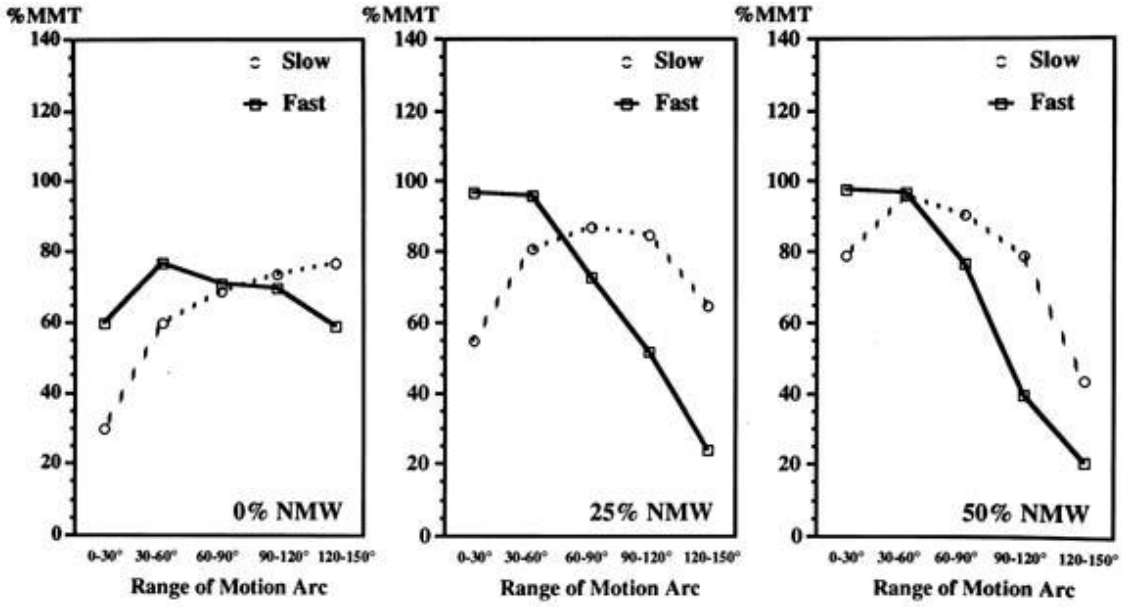

Figura 36: Effetto della velocità sul muscolo deltoide laterale nelle condizioni senza carico a sinistra, con 25% NMW al centro e 50% NMW a destra [127]

Per mitigare questo effetto è stata introdotta una funzione di compressione della dinamica (*warping function*) sigmoidale, applicata in fase di post-processing all'output del modello. Al fine di applicare la compressione in maniera controllata e indipendente dal range assoluto dei valori, l'output grezzo del modello viene preliminarmente normalizzato nell'intervallo [0, 1] utilizzando i parametri (minimo e massimo) del set di training, tramite *min-max scaling*. La funzione di *warping* è definita come:

$$y = \frac{1}{1 + e^{-K(x-Th)}} \tag{12}$$

dove x che rappresenta l'output del modello normalizzato, K è il coefficiente di pendenza (*slope*), che determina la rapidità della transizione e Th è la soglia (*threshold*), che definisce il punto di flesso della curva.

La selezione dei parametri ottimali è avvenuta tramite un'analisi empirica del comportamento della funzione nell'intervallo normalizzato. L'obiettivo era preservare la sensibilità per piccole escursioni attenuando al contempo le variazioni angolari medio-alte. Per ottenere questo effetto è stata sfruttata la seconda parte della sigmoide (dopo il cambiamento di pendenza), che con un'opportuna *K* è possibile ottenere un comportamento lineare e sfruttare il plateau per ottenere un effetto di saturazione nella parte finale del range. Tuttavia, per compiere un'effettiva attenuazione è necessario sottrarre il valore iniziale in quanto per definizione della sigmoide la seconda parte inizia a 0.5. Il fattore di attenuazione può essere modulato spostando opportunamente la *threshold* in un intorno dello zero. La formula finale della funzione di *warping* diventa:

$$y = \frac{1}{1 + e^{-K(x-Th)}} - \frac{1}{1 + e^{K(Th)}} \tag{13}$$

I valori individuati come idonei comprendono uno *slope* pari a 4 e una *threshold* posizionata leggermente in avanti rispetto allo zero (0.025), in modo da attivare la risposta angolare significativa solo al superamento di determinati livelli di confidenza del modello. La funzione è mostrata in Figura 37

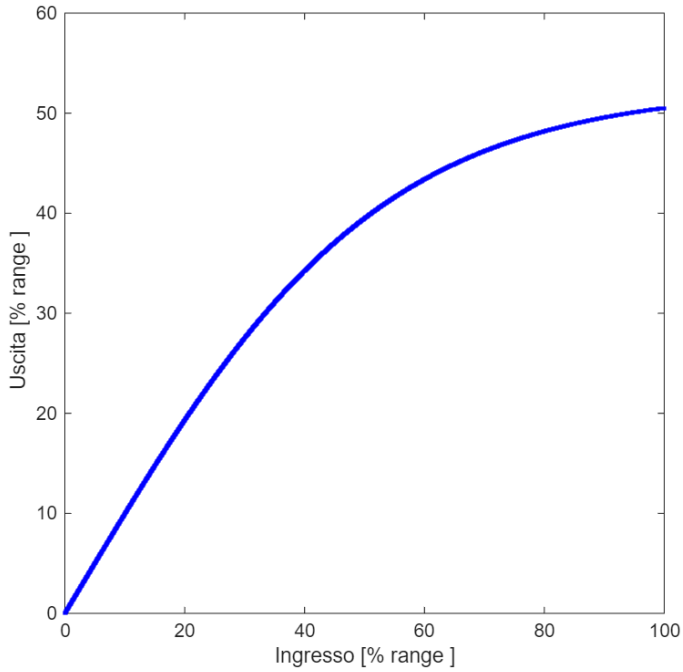

Figura 37: Rappresentazione grafica della funzione di warping sigmoidale in funzione della percentuale del range del modello

Il segnale in uscita viene in seguito de-normalizzato applicando la formula inversa *del min-max scaling:*

$$x_{denorm} = \mathrm{m} + (\mathrm{M} - \mathrm{m}) * y \tag{13}$$

con m e M rispettivamente il valore minimo e massimo del modello.

Analisi dei dati

Per quantificare l'accuratezza del modello nella stima dell'angolo di abduzione durante i task di *reaching*, sono stati calcolati diversi parametri per ogni singola ripetizione validata.

In primo luogo, è stata valutata l'escursione angolare complessiva (*Range Angolare*, RA). Per ogni i-esima ripetizione, il range è stato calcolato come la differenza tra il valore massimo e il valore minimo raggiunti durante la fase di andata, sia per l'angolo misurato dai sensori inerziali RA_{real} che per quello stimato dal modello RA_{stima}:

$$RA_i = max(\theta_i) - min(\theta_i) \tag{14}$$

Successivamente, è stato calcolato l'errore di stima del range ΔRA come differenza tra il range reale e quello ricostruito:

$$\Delta RA_{\mathrm{i}} = RA_{\mathrm{real,i}} - \mathrm{RA}_{\mathrm{stima,i}} \tag{15}$$

L'analisi statistica è stata condotta utilizzando un test non parametrico per ovviare alla ridotta numerosità del campione sperimentale. È stato scelto il *Wilcoxon rank-sum test*, in quanto rappresenta il metodo principale per il confronto di distribuzioni indipendenti qualora i presupposti di normalità non siano verificati, come nel caso in analisi.

In prima istanza, l'obiettivo è stato validare la capacità del modello nel discriminare le variazioni dell'angolo di abduzione nei vari task, in particolare tra i task 1 o 2, in cui avviene un *reaching* fisiologico, e il task 3 in cui sono simulati i movimenti compensatori. L'analisi statistica è stata condotta sulle variazioni angolari complessive individuate dal modello, verificando la presenza di differenze significative tra le distribuzioni stimate nelle diverse condizioni sperimentali. Successivamente, è stata valutata la capacità del modello nel quantificare il range angolare confrontando i dati stimati con i valori prodotti dall'IMU. Questa analisi non è stata condotta nell'ottica di ricostruire la cinematica, quanto piuttosto con l'obiettivo di fornire un biofeedback sui livelli di sovra-attivazione muscolare. L'intera validazione è stata infine condotta sia intra-soggetto, per valutare la coerenza del modello sul singolo individuo, sia su scala di popolazione, per generalizzare i risultati all'intero campione sperimentale.

Risultati e discussione

In questo capitolo vengono presentati i risultati ottenuti relativi alla creazione e validazione preliminare del modello di regressione proposto. L'analisi procede gerarchicamente: in primo luogo viene discussa la selezione delle ripetizioni valide e la loro segmentazione, successivamente, vengono valutate le metriche di bontà del fitting (*Goodness of Fit*) per le funzioni di fitting testate. Infine, vengono presentati i risultati della validazione del modello.

Selezione delle ripetizioni valide e segmentazione

L'applicazione dei criteri di validità cinematica ha comportato la selezione finale di 30 ripetizioni sulle 45 totali acquisite (tasso di accettazione del 66.7%). In Tabella 2 sono mostrati i risultati della classificazione delle tre ripetizioni per ciascun soggetto.

Tabella 2: Risultati dell'algoritmo di esclusione sui movimenti di abdo-adduzione utilizzati per il training del modello. La tabella illustra la classificazione in prove "valide" e "non valide" per ciascuna delle tre ripetizioni eseguite dai 15 soggetti sperimentali (Subj 01 - 15).

Soggetto	***Ripetizione 1***	***Ripetizione 2***	***Ripetizione 3***
Subj 01	valida	valida	valida
Subj 02	non valida	valida	valida
Subj 03	valida	valida	valida
Subj 04	non valida	valida	valida
Subj 05	valida	valida	non valida
Subj 06	valida	valida	valida
Subj 07	non valida	valida	valida
Subj 08	valida	valida	valida
Subj 09	valida	non valida	valida
Subj 10	valida	valida	non valida
Subj 11	non valida	non valida	valida
Subj 12	valida	valida	valida
Subj 13	non valida	non valida	non valida
Subj 14	valida	non valida	valida
Subj 15	non valida	non valida	non valida

L'analisi della distribuzione delle ripetizioni scartate ha evidenziato una certa variabilità inter-soggetto:

- 4 soggetti hanno mantenuto tutte le ripetizioni valide (esempio in Figura 38)
- 9 soggetti hanno subito un'esclusione parziale: per 8 di essi è stata scartata una sola ripetizione (esempio in Figura 39), mentre per un soggetto ne sono state scartate due.
- 2 soggetti sono stati interamente esclusi dall'analisi a causa della scarsa qualità dell'esecuzione in tutte le prove (esempio in Figura 40)

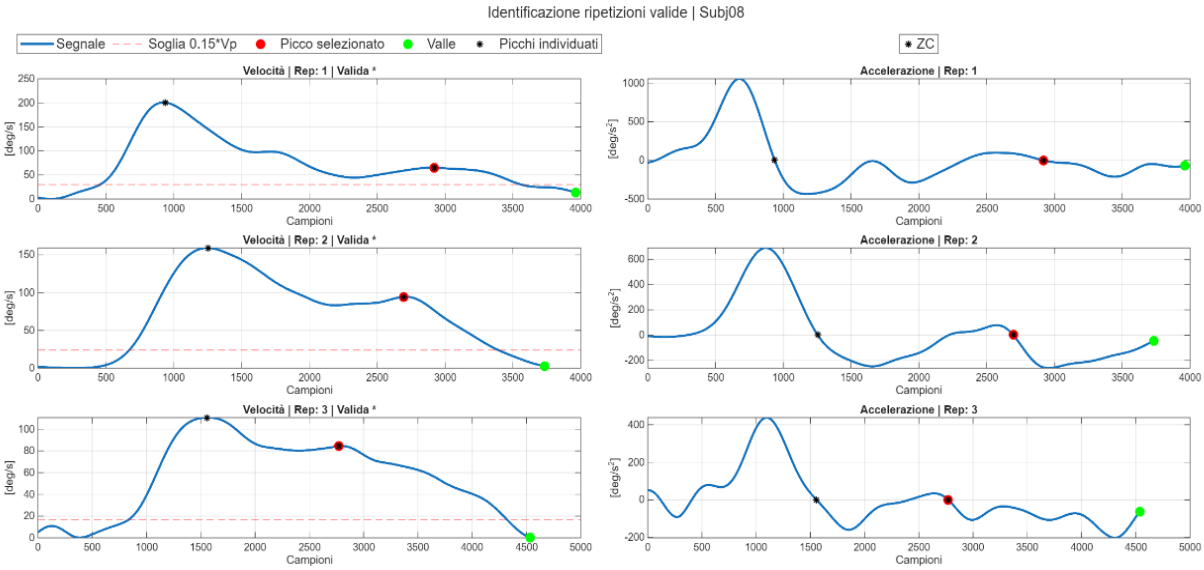

Figura 38. Visualizzazione dei criteri di inclusione/esclusione basati sulle derivate cinematiche su un soggetto ("Subj08") con tutte le ripetizioni valide. Ogni riga illustra una ripetizione: a sinistra il profilo di velocità, a destra quello di accelerazione. I punti evidenziati rappresentano gli eventi individuati dall'algoritmo (asterisco nero: picchi individuati, pallino rosso: picco selezionato, pallino verde valle successiva) con la relativa soglia di inclusione (linea tratteggiata colonna di sinistra).

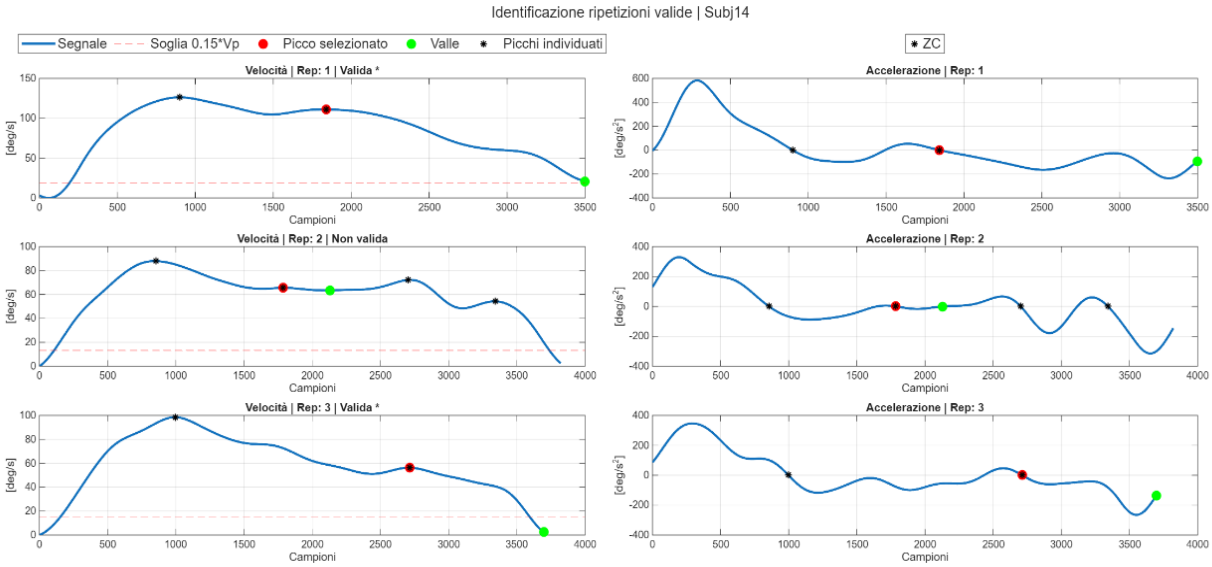

Figura 39: Visualizzazione dei criteri di inclusione/esclusione basati sulle derivate cinematiche su un soggetto ("Subj14") con esclusione parziale singola (ripetizione 2). Ogni riga illustra una ripetizione: a sinistra il profilo di velocità, a destra quello di accelerazione. I punti evidenziati rappresentano gli eventi individuati dall'algoritmo (asterisco nero: picchi individuati, pallino rosso: picco selezionato, pallino verde valle successiva) con la relativa soglia di inclusione (linea tratteggiata colonna di sinistra).

:

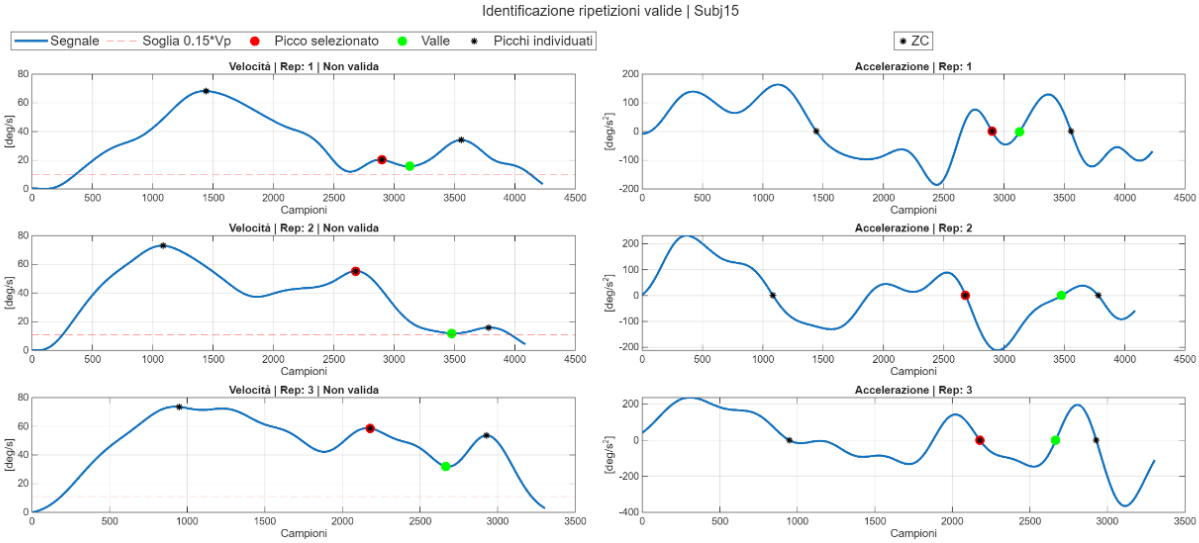

Figura 40: Visualizzazione dei criteri di inclusione/esclusione basati sulle derivate cinematiche su un soggetto ("Subj15") con nessuna ripetizione valida. Ogni riga illustra una ripetizione: a sinistra il profilo di velocità, a destra quello di accelerazione. I punti evidenziati rappresentano gli eventi individuati dall'algoritmo con la relativa soglia di inclusione (linea tratteggiata colonna di sinistra) e soglia dello zero-crossing (linea tratteggiata colonna di destra).

Dall'analisi dei grafici emerge come i soggetti mostrino una fase iniziale di movimento rapido, evidenziata dal picco di accelerazione, seguita da una successiva decelerazione e da una nuova fase accelerativa nella parte centrale del gesto. Nelle ripetizioni valide, la decelerazione finale avviene in modo costante e controllato; al contrario, in quelle non valide, si osserva una tendenza ad accelerare nuovamente, specialmente nella fase terminale del movimento. Questo fenomeno può essere ricondotto alla natura stessa del protocollo di calibrazione: dovendo riprodurre i movimenti di un avatar, la maggior parte dei soggetti non riesce a eseguire una ripetizione fluida, tendendo piuttosto a modulare la velocità per 'inseguire' il modello virtuale, in particolare nei punti di riferimento spaziale più facilmente individuabili, in assenza di un biofeedback dedicato

Per quanto concerne l'applicazione dell'algoritmo delle componenti non monotone, i risultati di alcuni soggetti campione sono presentati in Figura 41 e Figura 42.

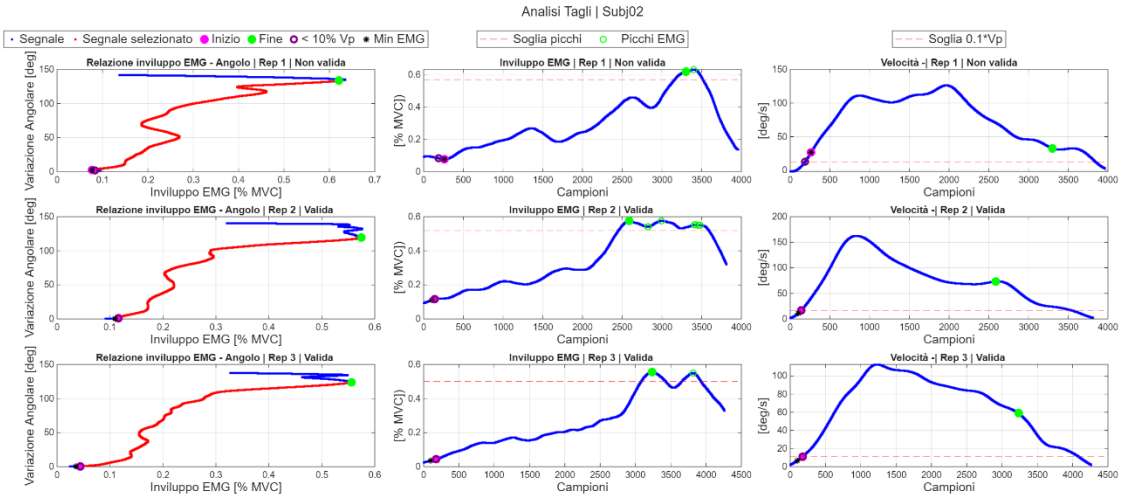

Figura 41 Esempio di identificazione degli istanti di inizio e fine movimento relativi a "Subj02". I grafici sono organizzati per righe (singole ripetizioni) e colonne (variabili analizzate): grafico inviluppo EMG-Angolo di abduzione (sinistra), inviluppo EMG (centro) e velocità angolare (destra).

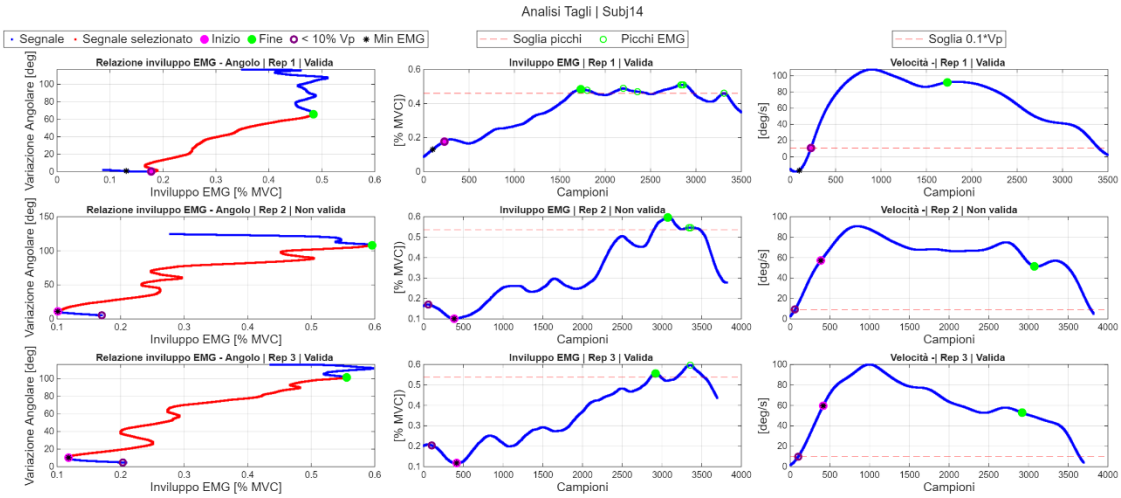

Figura 42: Esempio di identificazione degli istanti di inizio e fine movimento relativi a "Subj14". I grafici sono organizzati per righe (singole ripetizioni) e colonne (variabili analizzate): grafico inviluppo EMG-Angolo di abduzione (sinistra), inviluppo EMG (centro) e velocità angolare (destra).

Dall'ispezione visiva emerge come l'algoritmo di segmentazione attenui notevolmente le componenti non monotone, con particolare efficacia nella fase terminale dell'abduzione, sia per le ripetizioni classificate come valide che per quelle scartate.

Per quanto concerne la determinazione dell'istante iniziale, l'approccio ibrido ha dimostrato una duplice utilità: il vincolo sull'inviluppo EMG risulta determinante nella rimozione delle componenti di disturbo (come evidenziato in Figura 42, ripetizioni 2 e 3), mentre la condizione sulla velocità agisce efficacemente nell'eliminare le fasi di stasi tra una ripetizione e la successiva.

Nei casi ideali, in assenza di disturbi o pause prolungate, i due criteri convergono individuando istanti temporalmente molto vicini (Figura 41). Statisticamente, il criterio cinematico risulta predominante nel determinare l'avvio del movimento, attivandosi nel 70% dei casi considerando le sole ripetizioni valide.

Relativamente alla fase finale, si osserva in limitati casi la persistenza di moderate non-monotonie nel segnale individuato (Figura 41 ripetizione 1 e Figura 42 ripetizione 3). Tali irregolarità, attribuibili alla variabilità fisiologica del segnale elettromiografico, sono state volutamente conservate. L'imposizione di condizioni più stringenti comporterebbe una perdita eccessiva del contenuto informativo del segnale. Sarà compito della successiva fase di fitting approssimare e regolarizzare tali andamenti.

In termini quantitativi, il taglio finale risulta essere la condizione più onerosa, infatti l'algoritmo rimuove mediamente 19.24 ± 15.40 gradi (Figura 43.). Al contrario, il taglio iniziale ha un impatto minore sull'escursione articolare, rimuovendo 0.97 ± 2.52 gradi (Figura 43.). Complessivamente, considerando entrambi gli estremi, l'algoritmo esclude circa il 34% dei campioni totali acquisiti (Figura 43).

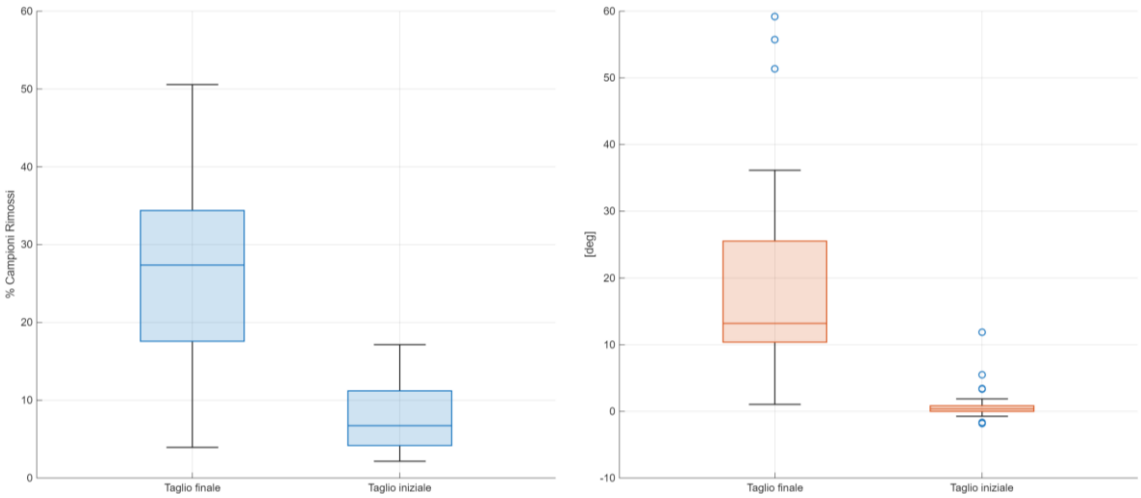

Figura 43: Quantificazione del segnale escluso dalla procedura di rimozione delle non-monotonie. Il grafico confronta l'impatto del taglio iniziale e finale in termini di durata temporale (campioni, a sinistra) e ampiezza cinematica (gradi, a destra), calcolati esclusivamente sulle ripetizioni valide.

Dall'analisi delle distribuzioni emerge una certa variabilità sia tra le ripetizioni eseguite dallo stesso soggetto che dalla popolazione, con casi che presentano un'esclusione piuttosto severa in termini angolari (50-60 gradi). Questo evidenzia come nonostante la natura temporizzata del task e l'applicazione del criterio di esclusione, si osservano sostanziali differenze di esecuzione tra i soggetti che accelerano e decelerano in fasi differenti del movimento, oltre alla naturale variabilità del segnale elettromiografico ampiamente documentata in letteratura. L'utilizzo dell'approccio *single trial* (una ripetizione per modello) permette di analizzare l'efficacia del modello anche nelle condizioni più drastiche, valutazione essenziale in un contesto di teleriabilitazione asincrona in cui il soggetto esegue la calibrazione e l'esecuzione dell'esercizio senza una supervisione costante.

Calibrazione del modello: scelta della funzione di fitting

I risultati delle metriche di valutazione sulla bontà dei fitting, calcolate su tutte le ripetizioni valide, sono riportate in Tabella 3 in termini di media e deviazione standard, mentre le distribuzioni dei dati sono mostrate in Figura 44 per l'Adjust-R^2 e in Figura 45 per RMSE.

Tabella 3: Confronto delle performance di fitting per le diverse funzioni matematiche testate. I valori riportati rappresentano la media e la deviazione standard dell'errore quadratico medio (RMSE) e del Adjust-R^2 calcolati sull'intero dataset di ripetizioni valide. L'ultima colonna indica il numero di parametri

Tipologia di fitting	***RMSE [deg] Media ± SD***	***Adj-R^2 Media ± SD***	***Num. Parametri***
Polinomiale (Grado 1)	12.7 ±5.2	0.82 ± 0.13	*2*
Polinomiale (Grado 2)	10.2 ± 4.3	0.88± 0.11	*3*
Polinomiale (Grado 3)	9.5 ± 4.5	0.89 ± 0.11	*4*
Polinomiale (Grado 4)	8.9 ± 4.6	0.90 ± 0.11	*5*
Esponenziale (2 termini)	10.2 ± 4.5	0.88 ± 0.11	*4*
Logistica (3 parametri)	9.2 ± 4.6	0.89 ± 0.12	*3*
Logistica (4 parametri)	9.1 ± 4.6	0.90 ± 0.11	*4*
Fourier (1 termine)	9.7 ± 4.4	0.88 ± 0.11	*4*
Fourier (2 termini)	8.7 ± 4.5	0.90 ± 0.11	*6*

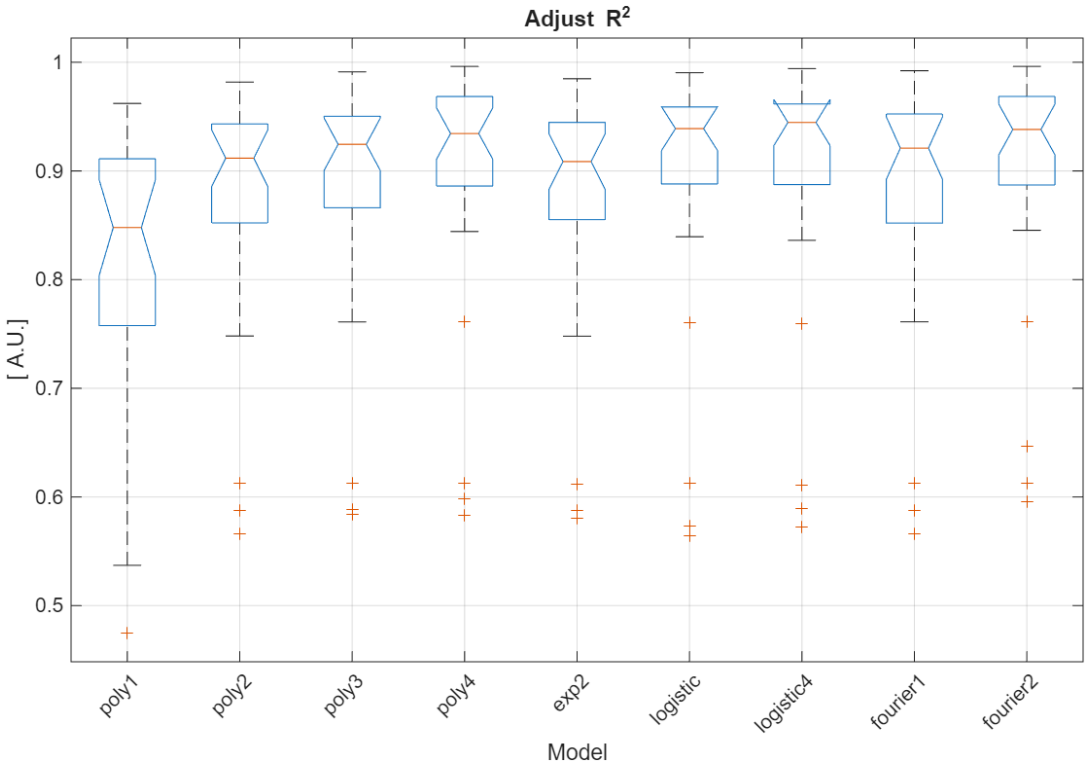

Figura 44: Distribuzione dei valori di Adjust-R² di tutte le funzioni di fitting testate, calcolate sull'intero dataset di ripetizioni valide

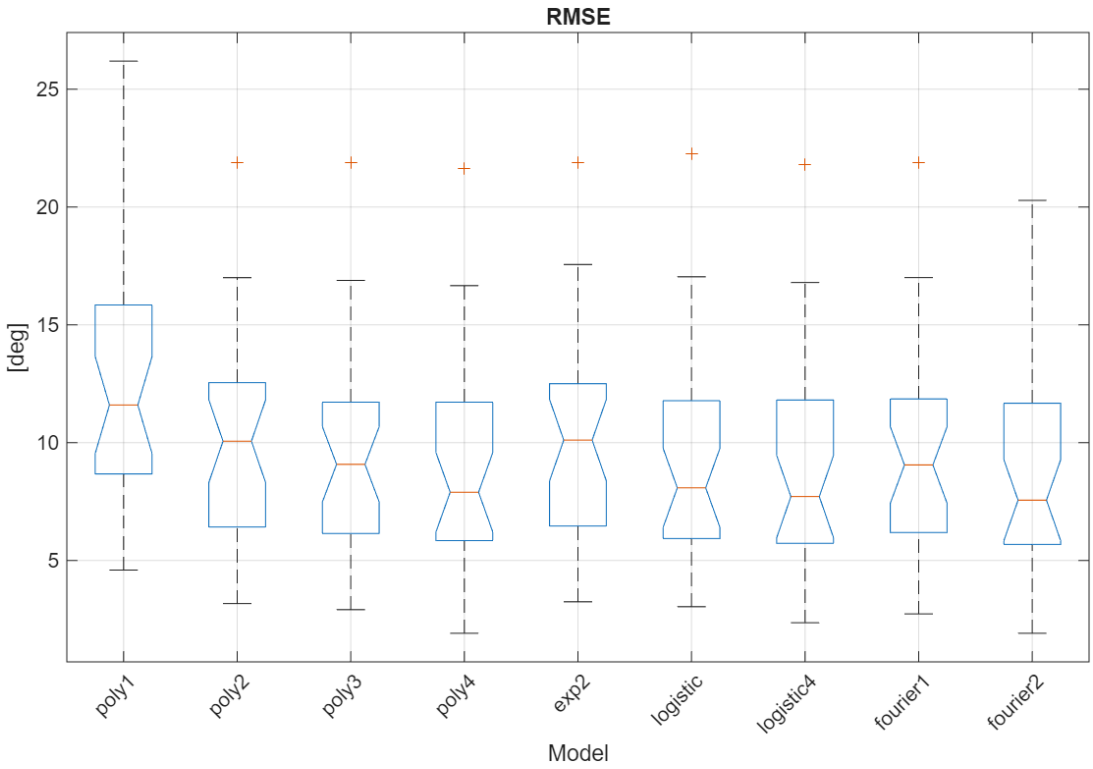

Figura 45: Distribuzione dei valori di RMSE di tutte le funzioni di fitting testate, calcolato sull'intero dataset di ripetizioni valide

L'analisi quantitativa ha evidenziato come le performance di fitting migliorino progressivamente all'aumentare della complessità del modello matematico. Le configurazioni migliori sono risultate il polinomio di ordine 4 (poly4), la funzione Logistica a 4 parametri (logistic4) e la funzione di Fourier a 2 termini (fourier2), con un errore medio (RMSE) rispettivamente di 8.9 ± 4.6, 9.1 ± 4.6 e 8.7 ± 4.5 gradi e un Adjust-R^2 confrontabile (Tabella 3).

La scelta finale è ricaduta sulla funzione logistica a 4 parametri, preferita ai modelli polinomiale di 4° ordine e di Fourier a due termini, sebbene questi ultimi mostrassero, in media, performance leggermente migliori sulla popolazione. La decisione è stata guidata da due motivazioni principali. In primo luogo, la logistica consente di ottenere un compromesso più favorevole tra complessità (numero di parametri) e accuratezza in quanto l'incremento prestazionale garantito dai modelli alternativi non è risultato sufficientemente rilevante. In secondo luogo, la funzione scelta presenta un comportamento più stabile agli estremi del range articolare. Infatti, le variazioni più evidenti tra i modelli si manifestano nella parte finale dell'abduzione: per variazioni repentine del segnale EMG, il polinomio e la funzione di Fourier tendono ad assumere un comportamento oscillatorio, mentre nella parte iniziale possono assumere valori negativi, come illustrato in Figura 46 e Figura 47. In aggiunta, il polinomio tende a divergere verso valori infiniti al di fuori dell'intervallo di interpolazione, rendendo la stima dell'angolo potenzialmente instabile. Il modello di Fourier invece, può risultare eccessivamente aderente alle fluttuazioni locali del segnale EMG, con il rischio di introdurre fenomeni di *overfitting*, senza apportare un reale miglioramento nella stima della variazione angolare.

Al contrario, la funzione logistica è caratterizzata da un andamento sigmoidale con asintoti orizzontali (saturazione), che rispecchia più fedelmente i limiti fisiologici dell'attivazione muscolare: il muscolo non può contrarsi infinitamente né generare angoli indefiniti, ma tende a un livello di massima attivazione (*plateau*). Inoltre, agendo opportunamente sui parametri di fitting, come il limite inferiore, è possibile vincolare la funzione ad un intervallo positivo, senza la necessità di implementare controlli o limitare ulteriormente il range in ingresso al modello.

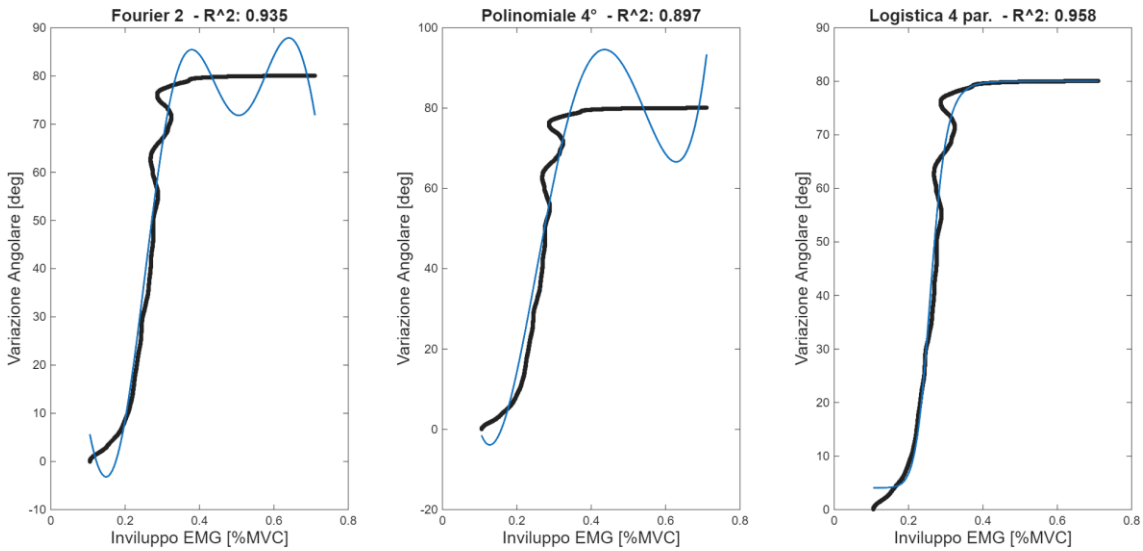

Figura 46: Confronto tra le funzioni di fitting: Fourier a 2 termini (a sinistra), Polinomiale di grado 4 (al centro), Logistica 4 parametri (a destra), e per "Subj01" prima ripetizione.

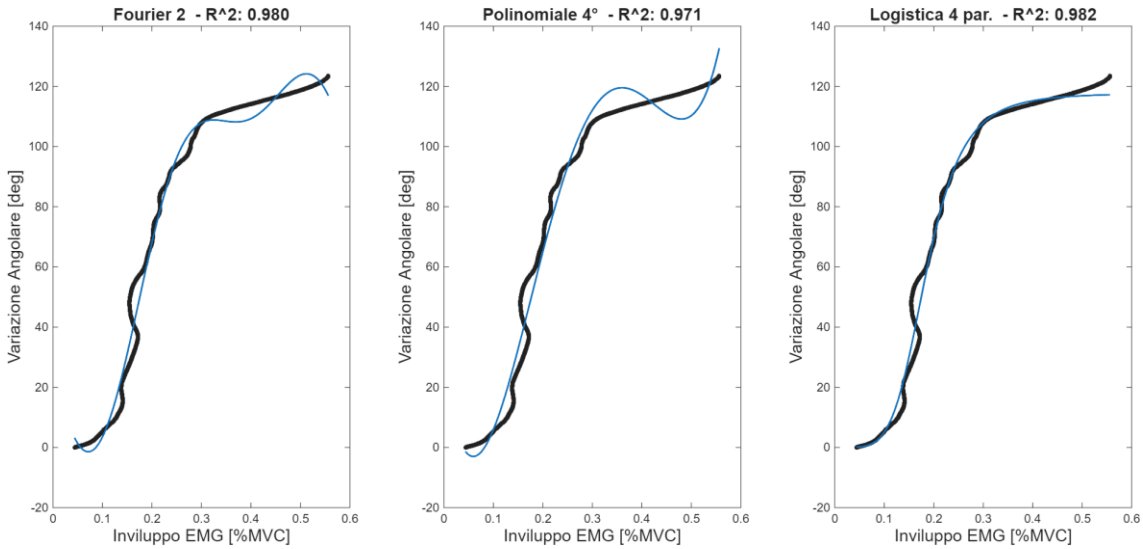

Figura 47: Confronto tra le funzioni di fitting: Fourier a 2 termini (a sinistra), Polinomiale di grado 4 (al centro), Logistica 4 parametri (a destra), e per "Subj02" terza ripetizione.

La funzione Logistica a 4 parametri rappresenta quindi il miglior compromesso tra accuratezza, numero di parametri e robustezza fisiologica, fornendo stime stabili e biologicamente plausibili anche nelle fasi estreme del movimento.

Risultati del modello

Per la validazione del modello sono stati esclusi dall'analisi i soggetti che presentavano le seguenti caratteristiche:

- Errori di ricostruzione della cinematica angolare sia nell'angolo di flesso-estensione orizzontale che nell'angolo di abdo-adduzione, in termini di angoli particolarmente rumorosi (Figura 48) e angoli non congruenti con la cinematica prevista dal protocollo nel task 3 (Figura 49).
- Qualità del fitting troppo bassa, valutata in termini di Adjust-R^2 inferiore a 0.7.

Le ripetizioni rimanenti a valle dei criteri sopra citati sono 13/30.

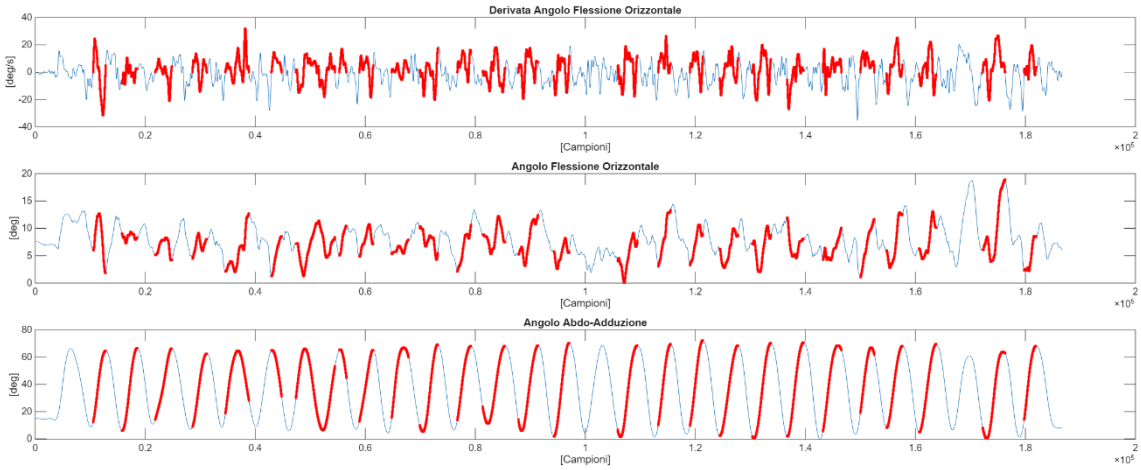

Figura 48: Esempio di esclusione per scarsa qualità del segnale cinematico (Subj 11,task 3). L'eccessiva rumorosità dell'angolo di flesso-estensione orizzontale compromette la corretta segmentazione delle fasi di andata (evidenziate in rosso). I pannelli mostrano, dall'alto verso il basso: la derivata dell'angolo di flessione (segnale di input per l'algoritmo di segmentazione), il profilo dell'angolo di flesso-estensione orizzontale e l'angolo di abdo-adduzione relativo al task 3.

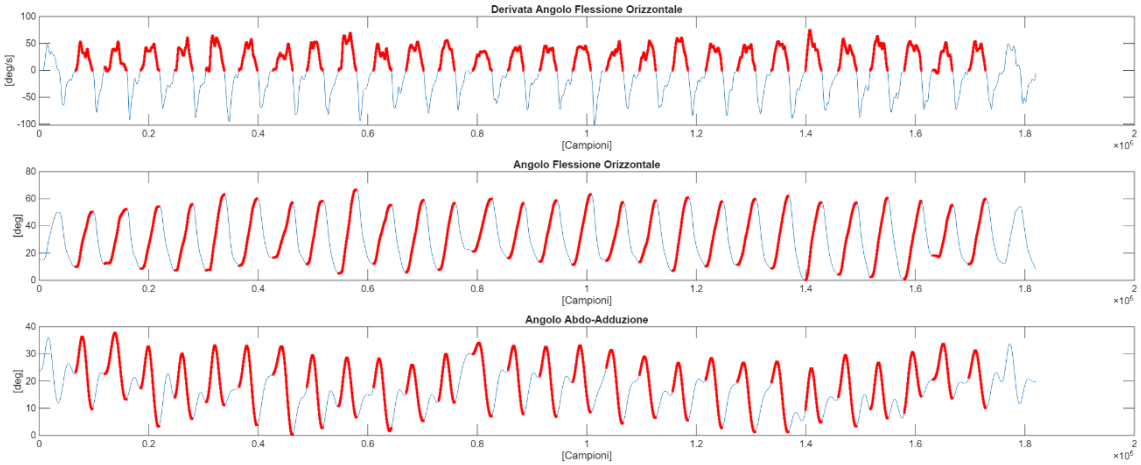

Figura 49 Esempio di esclusione per incongruenza cinematica rispetto al protocollo sperimentale (Subj10, task 3). I grafici riportano, in ordine discendente: la derivata dell'angolo di flessione utilizzata per la segmentazione, l'angolo di flesso-estensione orizzontale e l'angolo di abdo-adduzione. Le porzioni di segnale corrispondenti alle fasi di andata sono evidenziate in rosso *per ciascuna variabile.*

Alcuni esempi della variazione angolare stimata dal modello nelle tre condizioni sperimentali sono riportati Figura 50 (esempio di corretto comportamento, soggetto "*best case*") e in Figura 50 (esempio di comportamento non ottimale, soggetto "*worst case*").

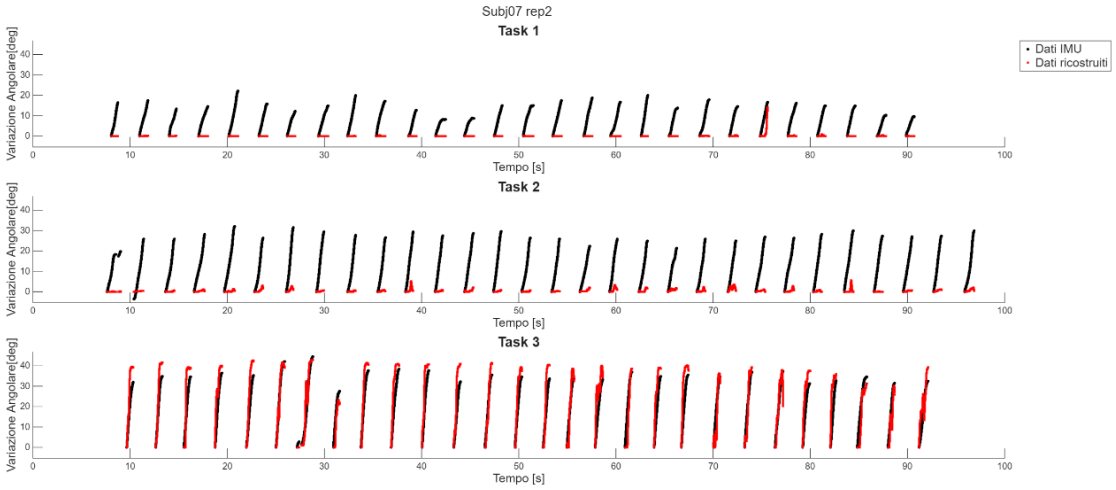

Figura 51: Variazione angolare di abduzione ricostruita dal modello durante i task di reaching in appoggio (task 1), senza appoggio (task 2) e con simulazione di movimenti compensatori (task 3) per il SUBJ07 ripetizione 2 (caso migliore). Per ogni grafico viene riportato: la variazione angolare estratta dai sensori IMU relativa al movimento di andata (in nero) e quella ricostruito dal modello (in rosso).

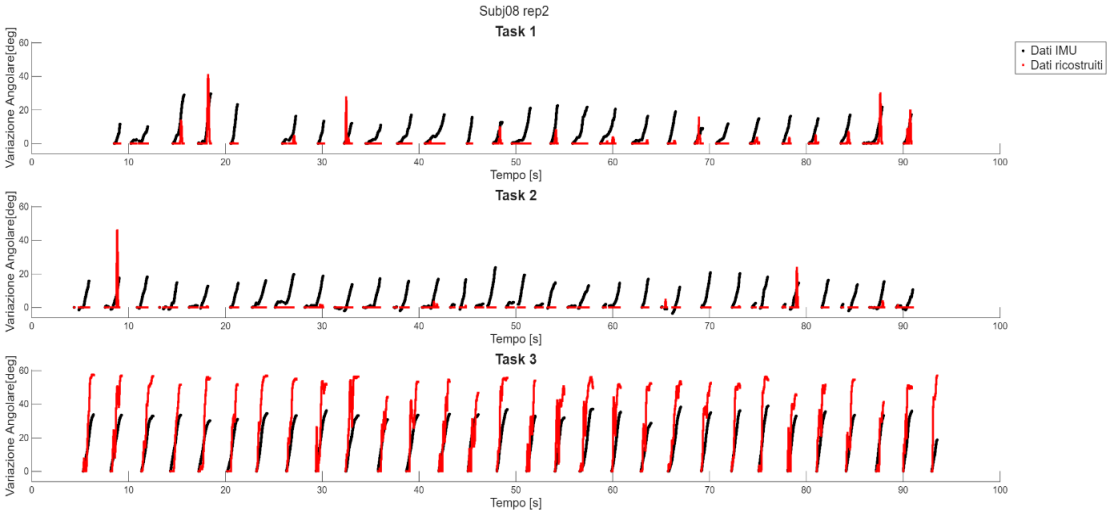

Figura 50: Variazione angolare di abduzione ricostruita dal modello durante i task di reaching in appoggio (task 1), senza appoggio (task 2) e con simulazione di movimenti compensatori (task 3) per il SUBJ08 ripetizione 2 (caso non ottimale). Per ogni grafico viene riportato: la variazione angolare estratta dai sensori IMU relativa al movimento di andata (in nero) e quella ricostruito dal modello (in rosso).

Dall'ispezione visiva delle ricostruzioni per il "*best case*" (Figura 50) emerge come il modello nei task 1 e 2, ovvero quelli relativi a *reaching* con e senza appoggio, tende a sopprimere quasi totalmente le variazioni angolari di abduzione e quando rilevate, risultano confrontabili con i dati provenienti dai sensori inerziali. Nel task 3 invece, nel quale vengono simulati i movimenti compensatori il modello è in grado di rilevarli correttamente, producendo una variazione angolare nettamente maggiore rispetto alle condizioni precedentemente descritte, e con una stima confrontabile con i dati inerziali per tutte le ripetizioni eseguite. D'altro canto, analizzando il "*worst case*" (Figura 50) è possibile notare che in questo caso il modello compie degli errori nella rilevazione dei compensi nel task 1 e 2, talvolta anche notevoli in termini di stima, ma comunque in maniera ancora limitata rispetto al numero totale di ripetizioni. La situazione più critica risiede nel task 3, dove sebbene il modello rilevi i compensi, le variazioni ricostruite sono nettamente maggiori rispetto a quelle estratte dai sensori inerziali con una sovra-stima sistematica del compenso. Analizzando lo stesso soggetto, ma con il modello costruito con un'altra ripetizione valida emerge un comportamento estremamente differente nei task, in particolare nei primi due si ripresenta una situazione ottimale, nella quale non vengono rilevate sovra-attivazioni, mentre nel task 3 l'entità della stima viene ridotta notevolmente con valori confrontabili con i dati IMU e in alcuni casi si verifica una sottostima. Questi risultati suggeriscono la presenza di una forte variabilità intra-soggetto nella stima del modello, determinata dall'esecuzione della calibrazione, nonostante i criteri di inclusione egli algoritmi di segmentazione introdotti.

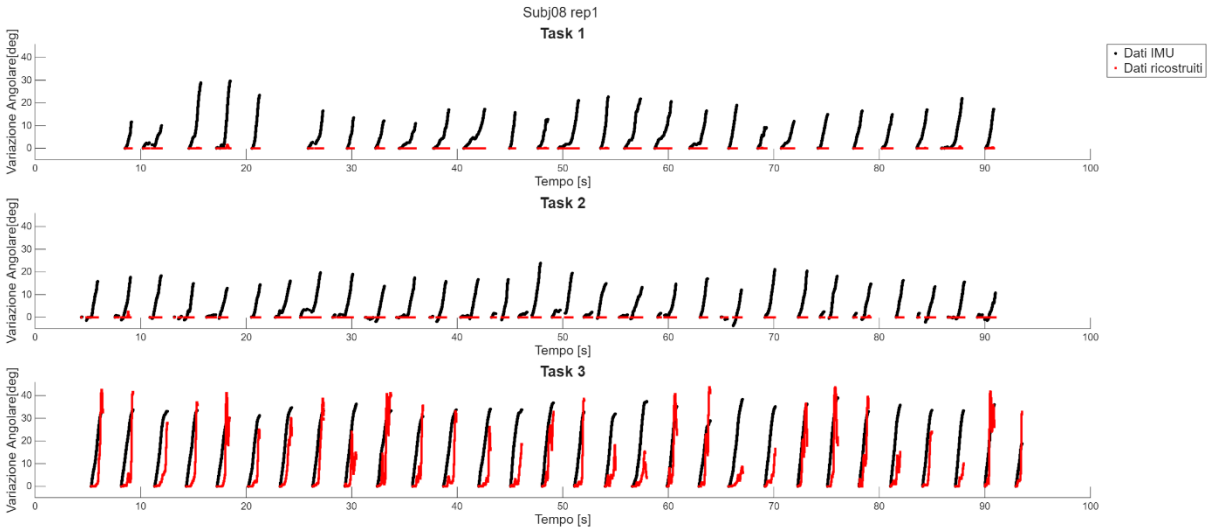

Figura 52: Variazione angolare di abduzione ricostruita dal modello durante i task di reaching in appoggio (task 1), senza appoggio (task 2) e con simulazione di movimenti compensatori (task 3) per il SUBJ08 ripetizione 1. Per ogni grafico viene riportato: la variazione angolare estratta dai sensori IMU relativa al movimento di andata (in nero) e quella ricostruito dal modello (in rosso).

I risultati dell'analisi sull'intera popolazione sono presentati nelle figure seguenti mediante boxplot, che consente di analizzare la distribuzione statistica delle metriche di interesse nei tre task motori. Nella Figura 53, viene mostrato il confronto delle distribuzioni dei dati reali (IMU) e quelli ricostruiti, in termini di range angolare di ogni ripetizione, mentre in Figura 54 sono riportate le distribuzioni degli errori commessi dal modello nella stima del range.

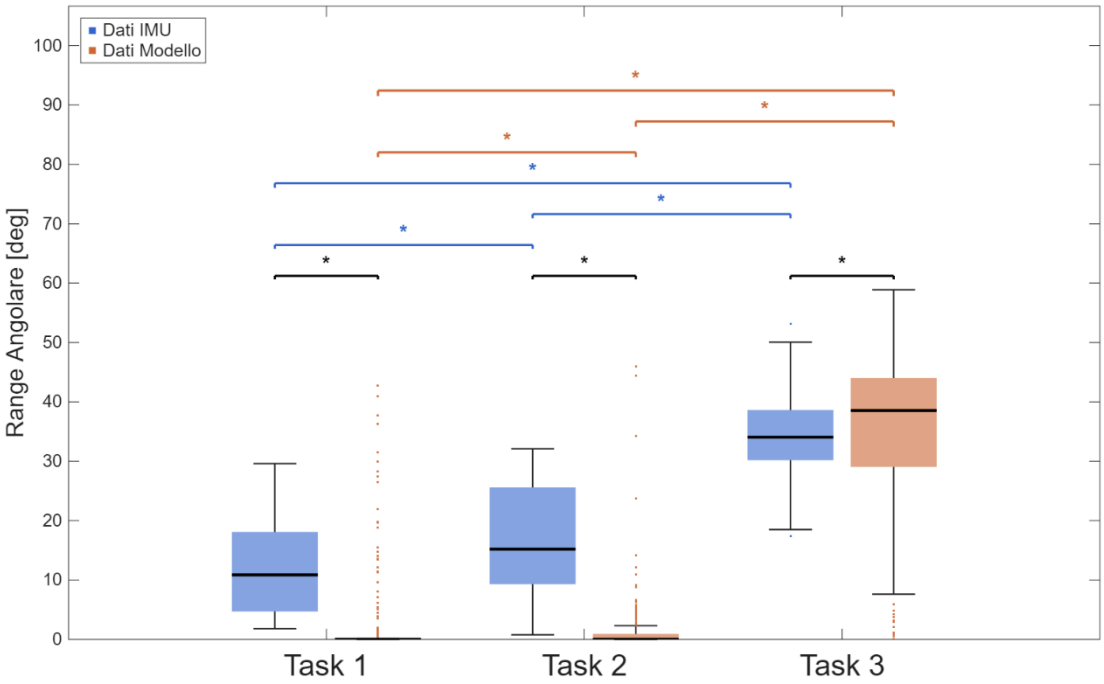

*Figura 53: Confronto tra le distribuzioni di Range Angolari dei dati IMU (box blu) e ricostruiti (box arancioni) per i tre Task. Le parentesi orizzontali indicano le differenze statisticamente significative rilevate dal test di Wilcoxon (con * $p < 0.01$),*

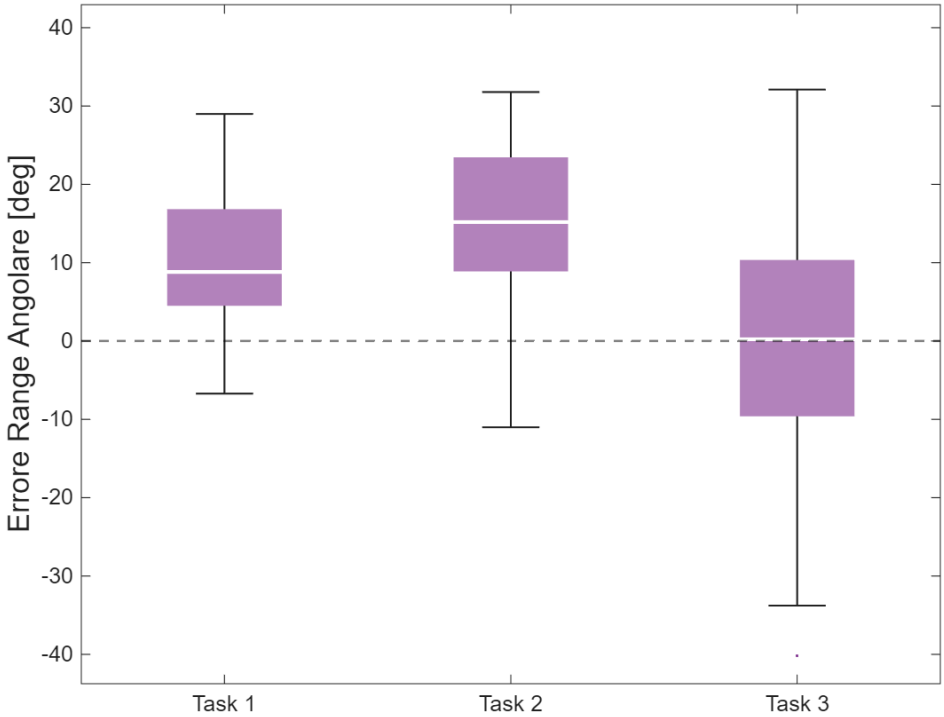

Figura 54: Analisi degli errori dei range angolari calcolati come differenza tra il dato reale (IMU) e quello ricostruito. I boxplot sono raggruppati per task (1, 2, 3) mentre la linea tratteggiata indica lo zero ideale (assenza di errore).

Tabella 4: Valori medi e deviazione standard del Range Angolare dei dati ricostruiti dai sensori inerziali (IMU) e prodotti dal modello per ogni ripetizione del dataset e relativo Errore di stima.

Task	**Metrica**	**IMU [gradi] (Media ± SD)**	**Modello [gradi] (Media ± SD)**
Task 1	Range	12 ± 8	3 ± 9
	Errore Range		9 ± 10
Task 2	Range	16 ± 9	2 ± 6
	Errore Range		14 ± 10
Task 3	Range	34. ± 7	35 ± 13
	Errore Range		-1 ± 12

In termini quantitativi (Tabella 4) nei task 1 e 2, caratterizzati da escursioni reali moderate (range medio rispettivamente di 11.89 ± 8 gradi e 16.05 ± 9 gradi), si osserva una sistematica sottostima della cinematica da parte del modello. L'angolo ricostruito risulta quasi nullo con un range angolare medio rispettivamente di 3 ± 9 gradi per il task 1 e 2 ± 6 gradi per il task 2, estendendo le evidenze dell'analisi soggetto-specifica all'intera popolazione, ovvero che per movimenti di piccola entità il sistema tende a sopprimere il segnale. Ciò è confermato dall'errore positivo in entrambe le condizioni (8.76 e 13.70 gradi), indicando che il movimento reale è costantemente superiore a quello predetto. Nonostante questi risultati, tale discrepanza non deve essere interpretata come un fallimento della stima. Al contrario, essa riflette il corretto comportamento selettivo del modello: progettato specificamente per isolare i movimenti compensatori il sistema tende giustamente a sopprimere o minimizzare l'output in presenza di gesti di *reaching* fisiologici (task 1 e 2), con differenze statisticamente significative (*p-value* $<$ 0.01) rispetto al task 3 nel quale sono presenti abduzioni anomale. La 'sottostima' conferma, dunque, l'elevata specificità del modello a minimizzare i falsi positivi durante l'esecuzione corretta del gesto.

Sebbene non sia l'obiettivo primario del modello è stata analizzata la capacità di quest'ultimo di stimare il range angolare durante le abduzioni compensatorie del task 3. Il dato più positivo emerge dall'osservazione dei valori medi: il modello ha stimato un range di abduzione di 35 gradi a fronte di un valore di 34 gradi misurato dal sistema IMU. Nonostante l'ottima accuratezza media, i risultati del Task 3 sono caratterizzati da un'elevata variabilità. La deviazione standard del modello (± 13 gradi) è quasi doppia rispetto a quella rilevata dall'IMU (± 7 gradi) riflettendosi in un errore di stima che oscilla sensibilmente (± 12 gradi) con un valore assoluto mediano di circa 7.5 gradi e una differenza statisticamente significativa tra le misure.

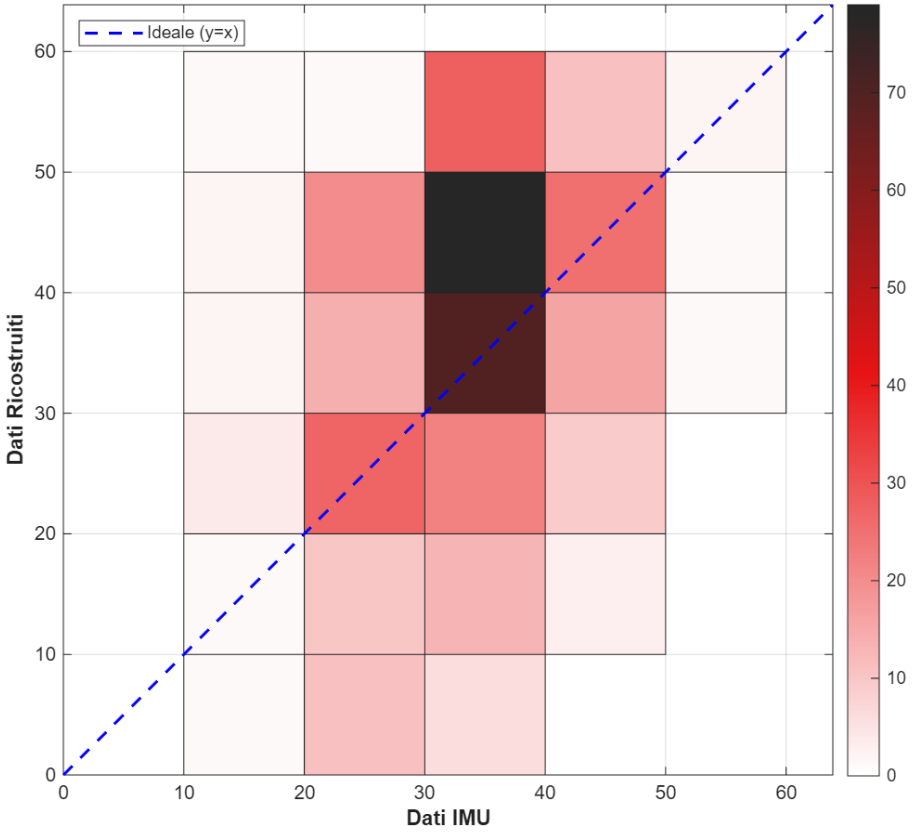

*Figura 55: I*stogramma di densità bidimensionale dei dati IMU e ricostruiti**.** *Il grafico mostra la distribuzione dei valori del range angolare di abduzione dei dati dell'IMU (asse delle ascisse) rispetto a quelli ricostruiti dal modello di regressione (asse delle ordinate). La linea tratteggiata rappresenta la bisettrice y = x (stima ideale). La colormap indica la densità dei campioni.*

Un aspetto fondamentale da sottolineare riguarda la finalità applicativa: il modello non è progettato per fornire una stima accurata e puntuale della variazione angolare, risultato difficilmente raggiungibile con l'approccio proposto. Piuttosto, nell'ottica di un'integrazione in un sistema di biofeedback, è più opportuno valutare la capacità del modello di fornire un'informazione qualitativa della performance basata su range (es. "corretto", "lieve compenso", "compenso eccessivo") risultando un'informazione più intuitiva rispetto a un dato numerico istantaneo soggetto a rumore.

Per verificare la capacità del modello di operare in tal senso, i dati sono stati raggruppati a intervalli (bin) di 10° (Figura 55). L'analisi dell'istogramma rivela che:

- Range 30-40°: È la zona dove si concentra la maggior parte dei dati reali (IMU). Coerentemente, il modello mostra la numerosità maggiore dei campioni in questo intervallo e in quello successivo (40-50°),
- Range 20-30° e 40-50°: Nonostante il ridotto numero di campioni rispetto al range precedente, la frequenza maggiore all'interno del bin corretto, tuttavia nell'intervallo 20-30° si registra la variabilità maggiore con variazioni stimate che ripercorrono un range ampio (0-50°), mentre per il bin 40-50°

- Range 10-20° e 50-60°: In questi intervalli si osserva una quantità esigua di dati (rispettivamente 13 e 4 ripetizioni).

In sintesi, i dati analizzati evidenziano come il punto di forza del modello risieda nella sua elevata capacità discriminativa: il sistema si è dimostrato estremamente affidabile nell'identificare l'insorgenza o l'assenza del movimento compensatorio. Per quanto riguarda invece la stima quantitativa dei range angolari, sebbene emerga una variabilità legata alla complessità del segnale sEMG e alla limitata numerosità del campione, i risultati ottenuti indicano un promettente punto di partenza per analisi e validazioni future.

Conclusioni

Il presente lavoro di tesi ha riguardato la progettazione e l'implementazione di un sistema di biofeedback integrato in un ambiente di realtà virtuale (VR), concepito per rispondere alle sfide della teleriabilitazione asincrona e guidare il paziente verso una corretta esecuzione del compito motorio. Il sistema fornisce all'utente informazioni dettagliate sulla cinematica del movimento in termini di estensione e velocità, integrando al contempo stimoli sonori volti a favorire la sincronizzazione e a ridurre il carico cognitivo durante l'esercizio. Il segnale elettromiografico (EMG) riveste un ruolo centrale, permettendo di valutare la correttezza del reclutamento muscolare e fornire un feedback sulle attivazioni anomale correlate a strategie motorie compensatorie.

Per l'analisi di tali parametri, si è resa necessaria l'integrazione della strumentazione VR (Meta Quest 3) e dei sensori per l'acquisizione del segnale elettromiografico di superficie (Due+). A tal proposito, è stata sviluppata una *pipeline* software dedicata alla gestione della comunicazione, dell'acquisizione, del pre-processing e del salvataggio dei dati. Tale architettura è stata progettata per adattarsi automaticamente sia a protocolli a sensore singolo che multi-sensore, operando sia in modalità PC-link che in modalità *stand-alone*.

La procedura di calibrazione permette di personalizzare l'esperienza riabilitativa in base alle capacità residue del singolo soggetto. Attraverso la stima del *Range of Motion* (ROM) residuo e della Massima Contrazione Volontaria (MVC), il sistema definisce soglie di attivazione e feedback che riflettono fedelmente le abilità motorie del partecipante.

In ottica di *gamification*, l'algoritmo di mapping sviluppato permette al partecipante di controllare in tempo reale il remo attraverso il movimento di *reaching*, garantendo l'esecuzione di una remata completa indipendentemente dal ROM residuo dell'utente e coerente con le fasi dell'esercizio riabilitativo.

Il conseguimento di tali obiettivi ha portato alla realizzazione di un sistema versatile, potenzialmente utilizzabile sia in ambito clinico e domiciliare per la riabilitazione, sia in ambito sportivo per l'allenamento e l'apprendimento del canottaggio sincronizzato.

Il principale sviluppo futuro di questa ricerca risiederà nella validazione del sistema di teleriabilitazione su campioni di soggetti sani e patologici, al fine di valutarne l'efficacia terapeutica, la sicurezza e l'usabilità attraverso l'ausilio di questionari standardizzati (*System Usability Scale* - SUS e *Simulator Sickness Questionnaire* - SSQ). I prossimi avanzamenti del sistema riguarderanno la generazione di report automatici per monitorare l'evoluzione della performance motoria nel tempo e l'integrazione di algoritmi adattivi (*Game Responsive Elements*) in grado di modulare dinamicamente il livello di difficoltà del compito per ottimizzare il processo di recupero e mantenere alto il livello di coinvolgimento.

Per quanto concerne il modello di regressione, i risultati preliminari dello studio sperimentale hanno evidenziato una buona accuratezza nell'identificare le sovra-attivazioni muscolari associate a movimenti di abduzione compensatoria. La peculiarità dell'approccio proposto risiede

nella capacità di adattarsi alle performance specifiche del singolo utente attraverso la fase di calibrazione iniziale, eliminando la necessità di impostare manualmente la soglia di attivazione. Grazie alla semplicità computazionale dell'architettura scelta, il modello si presta all'integrazione nel sistema di biofeedback, garantendone l'applicabilità in tempo reale. Inoltre, l'output del modello può essere impiegato per fornire un feedback visivo modulato su diversi livelli di intensità (ad esempio tramite una barra di performance) offrendo al paziente un'indicazione qualitativa sul grado di compenso attuato.

Sebbene l'obiettivo primario del modello non sia la stima puntuale dell'angolo articolare, l'analisi dei range angolari mostra come, mediamente, i dati stimati siano coerenti con quelli rilevati dai sensori inerziali (IMU). D'altro canto, è stata riscontrata una variabilità non trascurabile in alcuni soggetti, il che suggerisce la necessità di ricerche future volte ad ottenere movimenti più fluidi durante la calibrazione, aumentando il numero di ripetizioni valide. In alternativa, potrebbero essere esplorati approcci basati su relazioni medie, valutando il numero di ripetizioni ottimale per ottenere una stima più accurata e robusta. Il principale limite dello studio è rappresentato dalla numerosità ridotta del campione sperimentale. Risulta quindi necessario estendere la validazione a una coorte di soggetti più ampia, esplorando diverse strategie compensatorie di abduzione simulate. Tuttavia, il passaggio cruciale risiederà nella sperimentazione su soggetti patologici, nei quali i pattern di attivazione muscolare possono differire sensibilmente da quelli riscontrati nei soggetti sani.

Ringraziamenti

Un sentito ringraziamento va al Prof. Andrea Cereatti, per la fiducia accordatami e per aver reso possibile questo percorso di tesi in collaborazione esterna, e al Prof. Danilo Pani, per avermi accolto presso il suo laboratorio.

Un ringraziamento speciale va alla Dott. Giulia Sedda per il suo supporto e guida durante questo percorso e a tutti i componenti del MeDSP.

Infine, un ringraziamento immenso alla mia famiglia, non basterebbero le pagine di questa tesi per esprimere la mia gratitudine.

Grazie alla mia ragazza per avermi sempre sostenuto e creduto in me, dandomi la forza di superare anche i momenti più difficili, e agli amici, che hanno reso questo percorso indimenticabile arricchendolo di risate, avventure e di quella leggerezza necessaria per arrivare fin qui.

Bibliografia

[1] «WHO. A Health Telematics Policy in Support of WHO's Health-For-Allstrategy for Global Health Development Report of the WHO Group Consultation on Health Telematics. 11-16 December 1997, World Health Organization, Geneva, 1998. - References - Scientific Research Publishing». Disponibile su: https://www.scirp.org/reference/referencespapers?referenceid=2020379

[2] A. Cieza, K. Causey, K. Kamenov, S. W. Hanson, S. Chatterji, e T. Vos, «Global estimates of the need for rehabilitation based on the Global Burden of Disease study 2019: a systematic analysis for the Global Burden of Disease Study 2019», *The Lancet*, vol. 396, fasc. 10267, pp. 2006–2017, dic. 2020, doi: 10.1016/S0140-6736(20)32340-0.

[3] A. Peretti, F. Amenta, S. K. Tayebati, G. Nittari, e S. S. Mahdi, «Telerehabilitation: Review of the State-of-the-Art and Areas of Application», *JMIR Rehabilitation and Assistive Technologies*, vol. 4, fasc. 2, p. e7511, lug. 2017, doi: 10.2196/rehab.7511.

[4] D. Brennan *et al.*, «A Blueprint for Telerehabilitation Guidelines», *Int J Telerehabil*, vol. 2, fasc. 2, pp. 31–34, ott. 2010, doi: 10.5195/ijt.2010.6063.

[5] D. Hailey, R. Roine, A. Ohinmaa, e L. Dennett, «Evidence of benefit from telerehabilitation in routine care: a systematic review», *J Telemed Telecare*, vol. 17, fasc. 6, pp. 281–287, set. 2011, doi: 10.1258/jtt.2011.101208.

[6] M. McCue, A. Fairman, e M. Pramuka, «Enhancing Quality of Life through Telerehabilitation», *Physical Medicine and Rehabilitation Clinics*, vol. 21, fasc. 1, pp. 195–205, feb. 2010, doi: 10.1016/j.pmr.2009.07.005.

[7] S. Baffert, N. Hadouiri, C. Fabron, F. Burgy, A. Cassany, e G. Kemoun, «Economic Evaluation of Telerehabilitation: Systematic Literature Review of Cost-Utility Studies», *JMIR Rehabilitation and Assistive Technologies*, vol. 10, fasc. 1, p. e47172, set. 2023, doi: 10.2196/47172.

[8] M. T. Muñoz-Tomás *et al.*, «Telerehabilitation as a Therapeutic Exercise Tool versus Face-to-Face Physiotherapy: A Systematic Review», *International Journal of Environmental Research and Public Health*, vol. 20, fasc. 5, p. 4358, gen. 2023, doi: 10.3390/ijerph20054358.

[9] P. Seron *et al.*, «Effectiveness of Telerehabilitation in Physical Therapy: A Rapid Overview», *Physical Therapy*, vol. 101, fasc. 6, p. pzab053, giu. 2021, doi: 10.1093/ptj/pzab053.

[10] F. Velayati, H. Ayatollahi, e M. Hemmat, «A Systematic Review of the Effectiveness of Telerehabilitation Interventions for Therapeutic Purposes in the Elderly», *Methods Inf Med*, vol. 59, fasc. 2–03, pp. 104–109, mag. 2020, doi: 10.1055/s-0040-1713398.

[11] N. Aderinto, G. Olatunji, M. O. Abdulbasit, M. Edun, G. Aboderin, e E. Egbunu, «Exploring the efficacy of virtual reality-based rehabilitation in stroke: a narrative review of current evidence», *Ann Med*, vol. 55, fasc. 2, p. 2285907, 2023, doi: 10.1080/07853890.2023.2285907.

[12] P. Amorim, B. S. Santos, P. Dias, S. Silva, e H. Martins, «Serious Games for Stroke Telerehabilitation of Upper Limb - A Review for Future Research», *Int J Telerehabil*, vol. 12, fasc. 2, pp. 65–76, doi: 10.5195/ijt.2020.6326.

[13] S. H. H. Shah, A. S. T. Karlsen, M. Solberg, e I. A. Hameed, «A social VR-based collaborative exergame for rehabilitation: codesign, development and user study», *Virtual Reality*, vol. 27, fasc. 4, pp. 3403–3420, dic. 2023, doi: 10.1007/s10055-022-00721-8.

[14] D. M. M. Tonape, «GAINS AND CHALLENGES OF TELEREHABILITATION: A NARRATIVE REVIEW», *Indian Journal of Physical Therapy*, gen. 2025, doi: 10.63299/IJOPT.060329.

[15] «Benefits and drawbacks of telemedicine», *J Telemed Telecare*, vol. 11, fasc. 2, pp. 60–70, mar. 2005, doi: 10.1258/1357633053499886.

[16] M. Waller e C. Stotler, «Telemedicine: a Primer», *Curr Allergy Asthma Rep*, vol. 18, fasc. 10, p. 54, ago. 2018, doi: 10.1007/s11882-018-0808-4.
[17] M. Stoltzfus, A. Kaur, A. Chawla, V. Gupta, F. N. U. Anamika, e R. Jain, «The role of telemedicine in healthcare: an overview and update», *Egypt J Intern Med*, vol. 35, fasc. 1, p. 49, giu. 2023, doi: 10.1186/s43162-023-00234-z.
[18] I. Gareev *et al.*, «The opportunities and challenges of telemedicine during COVID-19 pandemic», *FBE*, vol. 13, fasc. 2, pp. 291–298, dic. 2021, doi: 10.52586/E885.
[19] D. Janela *et al.*, «Effectiveness of biofeedback-assisted asynchronous telerehabilitation in musculoskeletal care: A systematic review», *Digit Health*, vol. 9, p. 20552076231176696, 2023, doi: 10.1177/20552076231176696.
[20] O. M. Giggins, U. M. Persson, e B. Caulfield, «Biofeedback in rehabilitation», *J Neuroeng Rehabil*, vol. 10, p. 60, giu. 2013, doi: 10.1186/1743-0003-10-60.
[21] J. Parker, G. Mountain, e J. Hammerton, «A review of the evidence underpinning the use of visual and auditory feedback for computer technology in post-stroke upper-limb rehabilitation», *Disability and Rehabilitation: Assistive Technology*, vol. 6, fasc. 6, pp. 465–472, nov. 2011, doi: 10.3109/17483107.2011.556209.
[22] L. Brennan, E. Dorronzoro Zubiete, e B. Caulfield, «Feedback Design in Targeted Exercise Digital Biofeedback Systems for Home Rehabilitation: A Scoping Review», *Sensors*, vol. 20, fasc. 1, p. 181, gen. 2020, doi: 10.3390/s20010181.
[23] R. Sigrist, G. Rauter, R. Riener, e P. Wolf, «Augmented visual, auditory, haptic, and multimodal feedback in motor learning: A review», *Psychon Bull Rev*, vol. 20, fasc. 1, pp. 21–53, feb. 2013, doi: 10.3758/s13423-012-0333-8.
[24] S. Feng *et al.*, «EMG biofeedback combined with rehabilitation training may be the best physical therapy for improving upper limb motor function and relieving pain in patients with the post-stroke shoulder-hand syndrome: A Bayesian network meta-analysis», *Front. Neurol.*, vol. 13, gen. 2023, doi: 10.3389/fneur.2022.1056156.
[25] M. Doğan-Aslan, G. F. Nakipoğlu-Yüzer, A. Doğan, I. Karabay, e N. Özgirgin, «The effect of electromyographic biofeedback treatment in improving upper extremity functioning of patients with hemiplegic stroke», *J Stroke Cerebrovasc Dis*, vol. 21, fasc. 3, pp. 187–192, apr. 2012, doi: 10.1016/j.jstrokecerebrovasdis.2010.06.006.
[26] A. Sturma, L. A. Hruby, C. Prahm, J. A. Mayer, e O. C. Aszmann, «Rehabilitation of Upper Extremity Nerve Injuries Using Surface EMG Biofeedback: Protocols for Clinical Application», *Front. Neurosci.*, vol. 12, dic. 2018, doi: 10.3389/fnins.2018.00906.
[27] J. Spencer, S. L. Wolf, e T. M. Kesar, «Biofeedback for Post-stroke Gait Retraining: A Review of Current Evidence and Future Research Directions in the Context of Emerging Technologies», *Front. Neurol.*, vol. 12, mar. 2021, doi: 10.3389/fneur.2021.637199.
[28] M. Rogante, M. Grigioni, D. Cordella, e C. Giacomozzi, «Ten years of telerehabilitation: A literature overview of technologies and clinical applications», *NeuroRehabilitation*, vol. 27, fasc. 4, pp. 287–304, dic. 2010, doi: 10.3233/NRE-2010-0612.
[29] F. Almomani, A. A. Alghwiri, A. H. Alghadir, A. Al-Momani, e A. Iqbal, «Prevalence of upper limb pain and disability and its correlates with demographic and personal factors», *J Pain Res*, vol. 12, pp. 2691–2700, 2019, doi: 10.2147/JPR.S198480.
[30] K. Moulaei, A. Sheikhtaheri, M. S. Nezhad, A. Haghdoost, M. Gheysari, e K. Bahaadinbeigy, «Telerehabilitation for upper limb disabilities: a scoping review on functions, outcomes, and evaluation methods», *Arch Public Health*, vol. 80, fasc. 1, p. 196, ago. 2022, doi: 10.1186/s13690-022-00952-w.
[31] P. Rodrigues, C. Quaresma, M. Costa, F. Luz, e M. M. Fonseca, «Virtual Reality-Based Telerehabilitation for Upper Limb Recovery Post-Stroke: A Systematic Review of Design Principles, Monitoring, Safety, and Engagement Strategies», 12 gennaio 2025, *arXiv*: arXiv:2501.06899. doi: 10.48550/arXiv.2501.06899.
[32] M. F. Levin e M. Demers, «Motor learning in neurological rehabilitation», *Disability and Rehabilitation*, vol. 43, fasc. 24, pp. 3445–3453, nov. 2021, doi: 10.1080/09638288.2020.1752317.

[33] M. Hosseiniravandi, A. H. Kahlaee, H. Karim, L. Ghamkhar, e R. Safdari, «Home-based telerehabilitation software systems for remote supervising: a systematic review», *International Journal of Technology Assessment in Health Care*, vol. 36, fasc. 2, pp. 113–125, apr. 2020, doi: 10.1017/S0266462320000021.
[34] B. Nicolas, E. Leblong, B. Fraudet, P. Gallien, e P. Piette, «Telerehabilitation solutions in patient pathways: An overview of systematic reviews», *DIGITAL HEALTH*, vol. 10, p. 20552076241294110, set. 2024, doi: 10.1177/20552076241294110.
[35] L. R. Martin, S. L. Williams, K. B. Haskard, e M. R. Dimatteo, «The challenge of patient adherence», *Ther Clin Risk Manag*, vol. 1, fasc. 3, pp. 189–199, set. 2005.
[36] R. M. Ryan e E. L. Deci, «Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being», *Am Psychol*, vol. 55, fasc. 1, pp. 68–78, gen. 2000, doi: 10.1037//0003-066x.55.1.68.
[37] F. Pereira, S. Bermúdez I Badia, C. Jorge, e M. S. Cameirão, «The use of game modes to promote engagement and social involvement in multi-user serious games: a within-person randomized trial with stroke survivors», *J Neuroeng Rehabil*, vol. 18, fasc. 1, p. 62, apr. 2021, doi: 10.1186/s12984-021-00853-z.
[38] G. Burdea, V. Popescu, V. Hentz, e K. Colbert, «Virtual reality-based orthopedic telerehabilitation», *IEEE Transactions on Rehabilitation Engineering*, vol. 8, fasc. 3, pp. 430–432, set. 2000, doi: 10.1109/86.867886.
[39] L. E. Sucar, F. Orihuela-Espina, R. L. Velazquez, D. J. Reinkensmeyer, R. Leder, e J. Hernández-Franco, «Gesture Therapy: An Upper Limb Virtual Reality-Based Motor Rehabilitation Platform», *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 22, fasc. 3, pp. 634–643, mag. 2014, doi: 10.1109/TNSRE.2013.2293673.
[40] A. C. Alarcón-Aldana, M. Callejas-Cuervo, e A. P. L. Bo, «Upper Limb Physical Rehabilitation Using Serious Videogames and Motion Capture Systems: A Systematic Review», *Sensors*, vol. 20, fasc. 21, p. 5989, gen. 2020, doi: 10.3390/s20215989.
[41] N. Kato, T. Tanaka, S. Sugihara, e K. Shimizu, «Development and evaluation of a new telerehabilitation system based on VR technology using multisensory feedback for patients with stroke», *J Phys Ther Sci*, vol. 27, fasc. 10, pp. 3185–3190, ott. 2015, doi: 10.1589/jpts.27.3185.
[42] R. J. Adams *et al.*, «Virtual Activities of Daily Living for Recovery of Upper Extremity Motor Function», *IEEE Trans Neural Syst Rehabil Eng*, vol. 26, fasc. 1, pp. 252–260, gen. 2018, doi: 10.1109/TNSRE.2017.2771272.
[43] M. Trombetta, P. P. Bazzanello Henrique, M. R. Brum, E. L. Colussi, A. C. B. De Marchi, e R. Rieder, «Motion Rehab AVE 3D: A VR-based exergame for post-stroke rehabilitation», *Computer Methods and Programs in Biomedicine*, vol. 151, pp. 15–20, nov. 2017, doi: 10.1016/j.cmpb.2017.08.008.
[44] Z.-Q. Yang, D. Du, X.-Y. Wei, e R. K.-Y. Tong, «Augmented reality for stroke rehabilitation during COVID-19», *J Neuroeng Rehabil*, vol. 19, fasc. 1, p. 136, dic. 2022, doi: 10.1186/s12984-022-01100-9.
[45] C. Zhang, C. Greve, G. J. Verkerke, C. C. Roossien, H. Houdijk, e J. M. Hijmans, «Pilot Validation Study of Inertial Measurement Units and Markerless Methods for 3D Neck and Trunk Kinematics during a Simulated Surgery Task», *Sensors (Basel)*, vol. 22, fasc. 21, p. 8342, ott. 2022, doi: 10.3390/s22218342.
[46] A. Zedda *et al.*, «A User-Friendly Wearable Telerehabilitation System Based on Neuromotor Training for Mild Post-Stroke Patients: The DoMoMEA System», *IEEE Access*, vol. 13, pp. 122900–122913, 2025, doi: 10.1109/ACCESS.2025.3587614.
[47] E. Tsekleves, I. T. Paraskevopoulos, A. Warland, e C. Kilbride, «Development and preliminary evaluation of a novel low cost VR-based upper limb stroke rehabilitation platform using Wii technology», *Disabil Rehabil Assist Technol*, vol. 11, fasc. 5, pp. 413–422, 2016, doi: 10.3109/17483107.2014.981874.

[48] R. J. Adams, A. L. Ellington, K. Armstead, K. Sheffield, J. T. Patrie, e P. T. Diamond, «Upper Extremity Function Assessment Using a Glove Orthosis and Virtual Reality System», *OTJR (Thorofare N J)*, vol. 39, fasc. 2, pp. 81–89, apr. 2019, doi: 10.1177/1539449219829862.
[49] H. Ase *et al.*, «Effects of home-based virtual reality upper extremity rehabilitation in persons with chronic stroke: a randomized controlled trial», *J Neuroeng Rehabil*, vol. 22, fasc. 1, p. 20, feb. 2025, doi: 10.1186/s12984-025-01564-5.
[50] Q. Qiu *et al.*, «Development of the Home based Virtual Rehabilitation System (HoVRS) to remotely deliver an intense and customized upper extremity training», *J NeuroEngineering Rehabil*, vol. 17, fasc. 1, p. 155, nov. 2020, doi: 10.1186/s12984-020-00789-w.
[51] M. R. Donnelly *et al.*, «Acceptability of a Telerehabilitation Biofeedback System Among Stroke Survivors: A Qualitative Analysis», *OTJR (Thorofare N J)*, vol. 43, fasc. 3, pp. 549–557, lug. 2023, doi: 10.1177/15394492231153998.
[52] R. Llorens *et al.*, «Effectiveness of a combined transcranial direct current stimulation and virtual reality-based intervention on upper limb function in chronic individuals post-stroke with persistent severe hemiparesis: a randomized controlled trial», *J Neuroeng Rehabil*, vol. 18, fasc. 1, p. 108, lug. 2021, doi: 10.1186/s12984-021-00896-2.
[53] J. Hao, Z. He, X. Yu, e A. Remis, «Comparison of immersive and non-immersive virtual reality for upper extremity functional recovery in patients with stroke: a systematic review and network meta-analysis», *Neurol Sci*, vol. 44, fasc. 8, pp. 2679–2697, ago. 2023, doi: 10.1007/s10072-023-06742-8.
[54] J. P. Salisbury, T. M. Aronson, e T. J. Simon, «At-Home Self-Administration of an Immersive Virtual Reality Therapeutic Game for Post-Stroke Upper Limb Rehabilitation», in *Extended Abstracts of the 2020 Annual Symposium on Computer-Human Interaction in Play*, in CHI PLAY '20. New York, NY, USA: Association for Computing Machinery, nov. 2020, pp. 114–121. doi: 10.1145/3383668.3419935.
[55] K. Chatterjee, A. Buchanan, K. Cottrell, S. Hughes, T. W. Day, e N. W. John, «Immersive Virtual Reality for the Cognitive Rehabilitation of Stroke Survivors», *IEEE Trans Neural Syst Rehabil Eng*, vol. 30, pp. 719–728, 2022, doi: 10.1109/TNSRE.2022.3158731.
[56] M.-C. Juan, J. Elexpuru, P. Dias, B. S. Santos, e P. Amorim, «Immersive virtual reality for upper limb rehabilitation: comparing hand and controller interaction», *Virtual Reality*, vol. 27, fasc. 2, pp. 1157–1171, giu. 2023, doi: 10.1007/s10055-022-00722-7.
[57] S. Altukhaim, N. Sakabe, K. Nagaratnam, N. Mannava, T. Kondo, e Y. Hayashi, «Immersive virtual reality enhanced reinforcement induced physical therapy (EVEREST)», *Displays*, vol. 87, p. 102962, apr. 2025, doi: 10.1016/j.displa.2024.102962.
[58] D. B. Mekbib *et al.*, «A novel fully immersive virtual reality environment for upper extremity rehabilitation in patients with stroke», *Ann N Y Acad Sci*, vol. 1493, fasc. 1, pp. 75–89, giu. 2021, doi: 10.1111/nyas.14554.
[59] M. N. Ögün, R. Kurul, M. F. Yaşar, S. A. Turkoglu, Ş. Avci, e N. Yildiz, «Effect of Leap Motion-based 3D Immersive Virtual Reality Usage on Upper Extremity Function in Ischemic Stroke Patients», *Arq Neuropsiquiatr*, vol. 77, fasc. 10, pp. 681–688, 2019, doi: 10.1590/0004-282X20190129.
[60] G. Fregna, N. Schincaglia, A. Baroni, S. Straudi, e A. Casile, «A novel immersive virtual reality environment for the motor rehabilitation of stroke patients: A feasibility study», *Front Robot AI*, vol. 9, p. 906424, 2022, doi: 10.3389/frobt.2022.906424.
[61] F. Amin *et al.*, «Effectiveness of Immersive Virtual Reality-Based Hand Rehabilitation Games for Improving Hand Motor Functions in Subacute Stroke Patients», *IEEE Trans Neural Syst Rehabil Eng*, vol. 32, pp. 2060–2069, 2024, doi: 10.1109/TNSRE.2024.3405852.
[62] M. Benadduci *et al.*, «An Integrated Virtual Reality-Based Telerehabilitation Platform to Support Recovery and Maintenance of Functional Abilities Among Older Adults: Protocol for a Usability and Acceptability Study», *JMIR Res Protoc*, vol. 14, p. e68358, lug. 2025, doi: 10.2196/68358.

[63] J. Zhang *et al.*, «TouchMark: Partial Tactile Feedback Design for Upper Limb Rehabilitation in Virtual Reality», *IEEE Transactions on Visualization and Computer Graphics*, vol. 30, fasc. 11, pp. 7430–7440, nov. 2024, doi: 10.1109/TVCG.2024.3456173.
[64] S. Bae e H.-S. Park, «Development of Immersive Virtual Reality-Based Hand Rehabilitation System Using a Gesture-Controlled Rhythm Game With Vibrotactile Feedback: An fNIRS Pilot Study», *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 31, pp. 3732–3743, 2023, doi: 10.1109/TNSRE.2023.3312336.
[65] O. Ozioko e R. Dahiya, «Smart Tactile Gloves for Haptic Interaction, Communication, and Rehabilitation», *Advanced Intelligent Systems*, vol. 4, fasc. 2, p. 2100091, 2022, doi: 10.1002/aisy.202100091.
[66] M. Kim, C. Jeon, e J. Kim, «A Study on Immersion and Presence of a Portable Hand Haptic System for Immersive Virtual Reality», *Sensors*, vol. 17, fasc. 5, p. 1141, mag. 2017, doi: 10.3390/s17051141.
[67] I. Bortone *et al.*, «Wearable Haptics and Immersive Virtual Reality Rehabilitation Training in Children With Neuromotor Impairments», *IEEE Trans Neural Syst Rehabil Eng*, vol. 26, fasc. 7, pp. 1469–1478, lug. 2018, doi: 10.1109/TNSRE.2018.2846814.
[68] M. Lin, J. Huang, J. Fu, Y. Sun, e Q. Fang, «A VR-Based Motor Imagery Training System With EMG-Based Real-Time Feedback for Post-Stroke Rehabilitation», *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 31, pp. 1–10, 2023, doi: 10.1109/TNSRE.2022.3210258.
[69] L. A. Garcia-Villalba, A. G. Rodríguez-Ramírez, L. A. Rodríguez-Picón, L. C. Méndez-González, e S. M. Ghasemlou, «Interactive Platform for Hand Motor Rehabilitation Using Electromyography and Optical Tracking», *Applied Sciences*, vol. 15, fasc. 23, p. 12434, gen. 2025, doi: 10.3390/app152312434.
[70] X. Li, Z. Zhou, W. Liu, e M. Ji, «Wireless sEMG-based identification in a virtual reality environment», *Microelectronics Reliability*, vol. 98, pp. 78–85, lug. 2019, doi: 10.1016/j.microrel.2019.04.007.
[71] C.-H. Shih, P.-J. Lin, Y.-L. Chen, e S.-L. Chen, «A Post-Stroke Rehabilitation System With Compensatory Movement Detection Using Virtual Reality and Electroencephalogram Technologies», *IEEE Access*, vol. 12, pp. 61418–61432, 2024, doi: 10.1109/ACCESS.2024.3392513.
[72] K. O. Thielbar *et al.*, «Home-based Upper Extremity Stroke Therapy Using a Multiuser Virtual Reality Environment: A Randomized Trial», *Arch Phys Med Rehabil*, vol. 101, fasc. 2, pp. 196–203, feb. 2020, doi: 10.1016/j.apmr.2019.10.182.
[73] P. Raghavan, «Upper Limb Motor Impairment After Stroke», *Phys Med Rehabil Clin N Am*, vol. 26, fasc. 4, pp. 599–610, nov. 2015, doi: 10.1016/j.pmr.2015.06.008.
[74] S. Li, «Stroke Recovery Is a Journey: Prediction and Potentials of Motor Recovery after a Stroke from a Practical Perspective», *Life*, vol. 13, fasc. 10, p. 2061, ott. 2023, doi: 10.3390/life13102061.
[75] N. S. Ward, F. Brander, e K. Kelly, «Intensive upper limb neurorehabilitation in chronic stroke: outcomes from the Queen Square programme», *J Neurol Neurosurg Psychiatry*, vol. 90, fasc. 5, pp. 498–506, mag. 2019, doi: 10.1136/jnnp-2018-319954.
[76] P. H. McCrea, J. J. Eng, e A. J. Hodgson, «Biomechanics of reaching: clinical implications for individuals with acquired brain injury», *Disabil Rehabil*, vol. 24, fasc. 10, pp. 534–541, lug. 2002, doi: 10.1080/09638280110115393.
[77] M. C. Cirstea, A. B. Mitnitski, A. G. Feldman, e M. F. Levin, «Interjoint coordination dynamics during reaching in stroke», *Exp Brain Res*, vol. 151, fasc. 3, pp. 289–300, ago. 2003, doi: 10.1007/s00221-003-1438-0.
[78] J. V. G. Robertson, N. Roche, e A. Roby-Brami, «Influence of the side of brain damage on postural upper-limb control including the scapula in stroke patients», *Exp Brain Res*, vol. 218, fasc. 1, pp. 141–155, apr. 2012, doi: 10.1007/s00221-012-3014-y.

[79] M. F. Levin, J. A. Kleim, e S. L. Wolf, «What Do Motor "Recovery" and "Compensation" Mean in Patients Following Stroke?», *Neurorehabil Neural Repair*, vol. 23, fasc. 4, pp. 313–319, mag. 2009, doi: 10.1177/1545968308328727.
[80] M. C. Cirstea e M. F. Levin, «Compensatory strategies for reaching in stroke», *Brain*, vol. 123 (Pt 5), pp. 940–953, mag. 2000, doi: 10.1093/brain/123.5.940.
[81] M. F. Levin, D. G. Liebermann, Y. Parmet, e S. Berman, «Compensatory Versus Noncompensatory Shoulder Movements Used for Reaching in Stroke», *Neurorehabil Neural Repair*, vol. 30, fasc. 7, pp. 635–646, ago. 2016, doi: 10.1177/1545968315613863.
[82] Q. Xie, B. Sheng, J. Huang, Q. Zhang, e Y. Zhang, «A Pilot Study of Compensatory Strategies for Reach-to-Grasp-Pen in Patients with Stroke», *Appl Bionics Biomech*, vol. 2022, p. 6933043, 2022, doi: 10.1155/2022/6933043.
[83] A. Roby-Brami, S. Jacobs, N. Bennis, e M. F. Levin, «Hand orientation for grasping and arm joint rotation patterns in healthy subjects and hemiparetic stroke patients», *Brain Research*, vol. 969, fasc. 1, pp. 217–229, apr. 2003, doi: 10.1016/S0006-8993(03)02334-5.
[84] H. Ota *et al.*, «Movement Component Analysis of Reaching Strategies in Individuals With Stroke: Preliminary Study», *JMIR Rehabil Assist Technol*, vol. 10, p. e50571, dic. 2023, doi: 10.2196/50571.
[85] P. Lam *et al.*, «A haptic-robotic platform for upper-limb reaching stroke therapy: Preliminary design and evaluation results», *J NeuroEngineering Rehabil*, vol. 5, fasc. 1, p. 15, mag. 2008, doi: 10.1186/1743-0003-5-15.
[86] T. M. Sukal, M. D. Ellis, e J. P. A. Dewald, «Shoulder abduction-induced reductions in reaching work area following hemiparetic stroke: neuroscientific implications», *Exp Brain Res*, vol. 183, fasc. 2, pp. 215–223, nov. 2007, doi: 10.1007/s00221-007-1029-6.
[87] A. Roby-Brami, N. Jarrassé, e R. Parry, «Impairment and Compensation in Dexterous Upper-Limb Function After Stroke. From the Direct Consequences of Pyramidal Tract Lesions to Behavioral Involvement of Both Upper-Limbs in Daily Activities», *Front Hum Neurosci*, vol. 15, p. 662006, 2021, doi: 10.3389/fnhum.2021.662006.
[88] L. M. Pain, R. Baker, D. Richardson, e A. M. R. Agur, «Effect of trunk-restraint training on function and compensatory trunk, shoulder and elbow patterns during post-stroke reach: a systematic review», *Disabil Rehabil*, vol. 37, fasc. 7, pp. 553–562, 2015, doi: 10.3109/09638288.2014.932450.
[89] X. Wang, Y. Fu, B. Ye, J. Babineau, Y. Ding, e A. Mihailidis, «Technology-Based Compensation Assessment and Detection of Upper Extremity Activities of Stroke Survivors: Systematic Review», *J Med Internet Res*, vol. 24, fasc. 6, p. e34307, giu. 2022, doi: 10.2196/34307.
[90] M. Duff *et al.*, «An adaptive mixed reality training system for stroke rehabilitation», *IEEE Trans Neural Syst Rehabil Eng*, vol. 18, fasc. 5, pp. 531–541, ott. 2010, doi: 10.1109/TNSRE.2010.2055061.
[91] B. A. Valdés e H. F. M. Van der Loos, «Biofeedback vs. game scores for reducing trunk compensation after stroke: a randomized crossover trial», *Top Stroke Rehabil*, vol. 25, fasc. 2, pp. 96–113, mar. 2018, doi: 10.1080/10749357.2017.1394633.
[92] Z. Luo, A. E.-P. Lim, P. Durairaj, K. K. Tan, e V. Verawaty, «Development of a compensation-aware virtual rehabilitation system for upper extremity rehabilitation in community-dwelling older adults with stroke», *J Neuroeng Rehabil*, vol. 20, fasc. 1, p. 56, mag. 2023, doi: 10.1186/s12984-023-01183-y.
[93] A. Passon, T. Schauer, e T. Seel, «Inertial-Robotic Motion Tracking in End-Effector-Based Rehabilitation Robots», *Front Robot AI*, vol. 7, p. 554639, nov. 2020, doi: 10.3389/frobt.2020.554639.
[94] M. H. Foreman e J. R. Engsberg, «A virtual reality tool for measuring and shaping trunk compensation for persons with stroke: Design and initial feasibility testing», *J Rehabil Assist Technol Eng*, vol. 6, p. 2055668318823673, feb. 2019, doi: 10.1177/2055668318823673.

[95] J. L. Chen, S. Fujii, e G. Schlaug, «The use of augmented auditory feedback to improve arm reaching in stroke: a case series», *Disabil Rehabil*, vol. 38, fasc. 11, pp. 1115–1124, 2016, doi: 10.3109/09638288.2015.1076530.
[96] G. Nguyen, J. Maclean, e L. Stirling, «Quantification of Compensatory Torso Motion in Post-Stroke Patients Using Wearable Inertial Measurement Units», *IEEE Sensors Journal*, vol. 21, fasc. 13, pp. 15349–15360, lug. 2021, doi: 10.1109/JSEN.2021.3072010.
[97] Y. Fu *et al.*, «Vision-based Automatic Detection of Compensatory Postures of after-Stroke Patients During Upper-extremity Robot-assisted Rehabilitation: A Pilot Study in Reaching Movement», in *2020 International Conference on Assistive and Rehabilitation Technologies (iCareTech)*, ago. 2020, pp. 62–66. doi: 10.1109/iCareTech49914.2020.00019.
[98] J. Khoramdel, A. Moori, M. M. Moghaddam, e E. Najafi, «Compensatory Movement Detection in Upper Limb Rehabilitation With Deep Learning Methods», in *2021 9th RSI International Conference on Robotics and Mechatronics (ICRoM)*, nov. 2021, pp. 465–471. doi: 10.1109/ICRoM54204.2021.9663458.
[99] N. Nordin, S.-Q. Xie, e B. Wünsche, «Simple torso model for upper limb compensatory assessment after stroke», in *2016 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, lug. 2016, pp. 775–780. doi: 10.1109/AIM.2016.7576862.
[100] R. Ranganathan, R. Wang, B. Dong, e S. Biswas, «Identifying compensatory movement patterns in the upper extremity using a wearable sensor system», *Physiol Meas*, vol. 38, fasc. 12, pp. 2222–2234, nov. 2017, doi: 10.1088/1361-6579/aa9835.
[101] K. Ma *et al.*, «sEMG-Based Trunk Compensation Detection in Rehabilitation Training», *Front Neurosci*, vol. 13, p. 1250, 2019, doi: 10.3389/fnins.2019.01250.
[102] P. Xu, D. Xia, B. Zheng, L. Huang, e L. Xie, «A Novel Compensatory Motion Detection Method Using Multiple Signals and Machine Learning», *IEEE Sensors Journal*, vol. 22, fasc. 17, pp. 17162–17172, set. 2022, doi: 10.1109/JSEN.2022.3190503.
[103] G. Alankus e C. Kelleher, «Reducing compensatory motions in video games for stroke rehabilitation», in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, in CHI '12. New York, NY, USA: Association for Computing Machinery, mag. 2012, pp. 2049–2058. doi: 10.1145/2207676.2208354.
[104] S. Cai, X. Wei, E. Su, W. Wu, H. Zheng, e L. Xie, «Online compensation detecting for real-time reduction of compensatory motions during reaching: a pilot study with stroke survivors», *J Neuroeng Rehabil*, vol. 17, fasc. 1, p. 58, apr. 2020, doi: 10.1186/s12984-020-00687-1.
[105] S.-H. Lee e W.-K. Song, «Mitigating Trunk Compensatory Movements in Post-Stroke Survivors through Visual Feedback during Robotic-Assisted Arm Reaching Exercises», *Sensors*, vol. 24, fasc. 11, p. 3331, gen. 2024, doi: 10.3390/s24113331.
[106] C. Lirio-Romero *et al.*, «Electromyographic biofeedback improves upper extremity function: a randomized, single-blinded, controlled trial», *Physiotherapy*, vol. 110, pp. 54–62, mar. 2021, doi: 10.1016/j.physio.2020.02.002.
[107] N. Hesam-Shariati, T. Trinh, A. G. Thompson-Butel, C. T. Shiner, e P. A. McNulty, «A Longitudinal Electromyography Study of Complex Movements in Poststroke Therapy. 2: Changes in Coordinated Muscle Activation», *Front. Neurol.*, vol. 8, lug. 2017, doi: 10.3389/fneur.2017.00277.
[108] «Meta Quest 3: visore per la realtà virtuale di nuova generazione». [Online]. Disponibile su: https://www.meta.com/it/quest/quest-3/
[109] «Design immersive experiences». [Online]. Disponibile su: https://developers.meta.com/horizon/design/
[110] OT Bioelettronica s.r.l., «Due+Pro Manuale Utente v.2.1». OT Bioelettronica s.r.l., Torino, Italia, gennaio 2024. [Online]. Disponibile su: https://otbioelettronica.it/downloads
[111] OT Bioelettronica s.r.l., «SyncStation TCP Communication Protocol v2.8». OT Bioelettronica s.r.l., Torino, Italy, 10 dicembre 2024. [Online]. Disponibile su: https://www.otbioelettronica.it/it/download

[112] C.-C. Tseng e S.-C. Pei, «Stable IIR notch filter design with optimal pole placement», *IEEE Transactions on Signal Processing*, vol. 49, fasc. 11, pp. 2673–2681, nov. 2001, doi: 10.1109/78.960414.

[113] A. Tonon, R. Pozzo, S. Pollastri, R. Centemeri, e L. Mochizuki, «Analysis of Muscle Recruitment Patterns During Shoulder Rehabilitation Exercises With and Without Elastic Resistance», *J Musculoskelet Neuronal Interact*, vol. 25, fasc. 2, pp. 160–171, 2025, doi: 10.22540/JMNI-25-160.

[114] M. B. I. Reaz, M. S. Hussain, e F. Mohd-Yasin, «Techniques of EMG signal analysis: detection, processing, classification and applications», *Biol. Proced. Online*, vol. 8, fasc. 1, pp. 11–35, dic. 2006, doi: 10.1251/bpo115.

[115] S. Al-Qaisi e F. Aghazadeh, «Electromyography Analysis: Comparison of Maximum Voluntary Contraction Methods for Anterior Deltoid and Trapezius Muscles», *Procedia Manufacturing*, vol. 3, pp. 4578–4583, gen. 2015, doi: 10.1016/j.promfg.2015.07.475.

[116] A. Kurtoglu, R. Ciftci, B. Car, e N. Konar, «Investigation of Deltoid Muscle Activation From Different Angles in Body Building Athletes», *Sisli Etfal Hastan Tip Bul*, vol. 57, fasc. 3, pp. 410–415, set. 2023, doi: 10.14744/SEMB.2023.09522.

[117] C. J. De Luca, «The Use of Surface Electromyography in Biomechanics», *Journal of Applied Biomechanics*, vol. 13, fasc. 2, pp. 135–163, mag. 1997, doi: 10.1123/jab.13.2.135.

[118] «EMG Standards», International Society of Electrophysiology and Kinesiology (ISEK). Disponibile su: https://isek.org/emg-standards/

[119] A. Mark Burden, S. E. Lewis, e E. Willcox, «The effect of manipulating root mean square window length and overlap on reliability, inter-individual variability, statistical significance and clinical relevance of electromyograms», *Manual Therapy*, vol. 19, fasc. 6, pp. 595–601, dic. 2014, doi: 10.1016/j.math.2014.06.003.

[120] Abraham. Savitzky e M. J. E. Golay, «Smoothing and Differentiation of Data by Simplified Least Squares Procedures.», *Anal. Chem.*, vol. 36, fasc. 8, pp. 1627–1639, lug. 1964, doi: 10.1021/ac60214a047.

[121] N. T. Antony e P. J. Keir, «Effects of posture, movement and hand load on shoulder muscle activity», *Journal of Electromyography and Kinesiology*, vol. 20, fasc. 2, pp. 191–198, apr. 2010, doi: 10.1016/j.jelekin.2009.04.010.

[122] L. Botzer e A. Karniel, «A Simple and Accurate onset Detection Method for a Measured Bell-shaped Speed Profile», *Front Neurosci*, vol. 3, p. 61, giu. 2009, doi: 10.3389/neuro.20.002.2009.

[123] J. Wickham, T. Pizzari, K. Stansfeld, A. Burnside, e L. Watson, «Quantifying "normal" shoulder muscle activity during abduction», *J Electromyogr Kinesiol*, vol. 20, fasc. 2, pp. 212–222, apr. 2010, doi: 10.1016/j.jelekin.2009.06.004.

[124] H. Motulsky e A. Christopoulos, *Fitting Models to Biological Data Using Linear and Nonlinear Regression: A Practical Guide to Curve Fitting*. Oxford University Press, 2004.

[125] «Evaluating Goodness of Fit - MATLAB & Simulink». Disponibile su: https://it.mathworks.com/help/curvefit/evaluating-goodness-of-fit.html

[126] K. H. E. Beange, A. D. C. Chan, e R. B. Graham, «Sources of error during inertial sensing of human movement: a critical review of the fundamentals», in *2023 IEEE International Symposium on Medical Measurements and Applications (MeMeA)*, giu. 2023, pp. 1–6. doi: 10.1109/MeMeA57477.2023.10171885.

[127] S. W. Alpert, M. M. Pink, F. W. Jobe, P. J. McMahon, e W. Mathiyakom, «Electromyographic analysis of deltoid and rotator cuff function under varying loads and speeds», *Journal of Shoulder and Elbow Surgery*, vol. 9, fasc. 1, pp. 47–58, gen. 2000, doi: 10.1016/S1058-2746(00)90009-0.