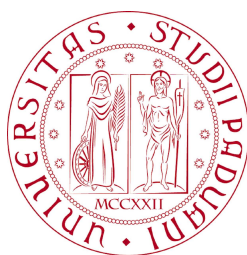


UNIVERSITÀ DEGLI STUDI DI PADOVA
FACOLTÀ DI INGEGNERIA



Finito di scrivere il giorno 21 Febbraio 2011 utilizzando L^AT_EX

UNIVERSITÀ DEGLI STUDI DI PADOVA
FACOLTÀ DI INGEGNERIA

—
DIPARTIMENTO DI INNOVAZIONE MECCANICA E GESTIONALE
—

TESI DI LAUREA TRIENNALE IN INGEGNERIA BIOMEDICA

INFLUENZA DEL FEEDBACK
SONORO SULLA FORMAZIONE DI
MODELLI INTERNI
NELL'ESCUZIONE DI TASK MOTORI

RELATORE: CH.MO PROF. ING. GIULIO ROSATI

LAUREANDO: MATTEO MILANI

ANNO ACCADEMICO 2010-2011

a mio padre, a mia sorella e a colei che mi ha insegnato a non arrendermi mai...

“ Ci son persone che sanno tutto e purtroppo è tutto quello che sanno ”

OSCAR WILDE

Indice

Sommario	XIII
Introduzione	XV
1 Revisione della Letteratura	1
1.1 Il ruolo della robotica riabilitativa	1
1.2 Apprendimento Motorio	10
1.2.1 Motor Control	11
1.2.2 Motor Learning	11
1.2.3 Feedback sonoro	12
1.3 Obiettivi del lavoro	14
1.3.1 Software sviluppato: Accenni	14
1.3.2 Test sui soggetti	15
2 Device sviluppato	19
2.1 Architettura del sistema	19
2.2 Implementazione dell'interfaccia GUI	21
2.2.1 Fattore di scalatura	24
2.2.2 Aggiornamento delle variabili: Funzione Draw	25
2.3 Utilizzo dell'interfaccia GUI	30
2.4 Joystick a due gradi di libertà	34
2.4.1 Simulink	37
2.5 Feedback audio	40
2.5.1 Regole di creazione dei pacchetti - protocollo OSC	40
2.5.2 Pacchetti creati dall'interfaccia	42

2.5.3	Invio dei pacchetti	43
2.5.4	Software di generazione di suoni: Pure Data	44
3	Test Sperimentali	47
3.1	Test Precedenti	47
3.2	Descrizione del test Variable Amplitude	49
3.2.1	Fase di Pre_Test	50
3.2.2	Fase di Post_Test	51
3.2.3	Fase di Var_Test	52
3.3	Tipologie di feedback audio applicati	54
3.3.1	Modifiche apportate ai suoni scelti	54
3.4	Soggetti reclutati per il test	56
3.4.1	Questionario	57
3.4.2	Gruppi ottenuti mediante randomizzazione	58
3.4.3	Impugnature	59
3.5	Problemi riscontrati	60
4	Analisi Statistica	67
4.1	Parametri calcolati per l'elaborazione	67
4.1.1	Distanza media dal target (dism)	68
4.1.2	Integrale del modulo della velocità relativa (dtt)	68
4.1.3	Ritardo medio (emx)	69
4.1.4	Errore medio di posizione in y (emy)	70
4.1.5	Errore quadratico medio in x (eqmx)	70
4.1.6	Errore quadratico medio in y (eqmy)	71
4.1.7	Lead error (lead)	71
4.1.8	Lag error (lag)	71
4.2	Analisi statistica	72
4.2.1	Test di Mann-Whitney	72
4.2.2	Unpaired t-test	73
4.2.3	Unpaired t-test con Welch's correction	73
4.3	Software per l'analisi statistica: GraphPad Prism	74
4.3.1	Procedimento	75

4.4	Risultati	76
4.4.1	Distanza dal target: dtt	76
4.4.2	Distanza media dal target: distm	77
4.4.3	Errore medio in x: emx	79
4.4.4	Confronto tra lead error e lag error	81
4.4.5	Errore medio (emy) ed errore quadratico medio in y (eqmy)	82
4.4.6	Riassunto dei risultati legati al Var_Test	86
	Conclusioni	87
	Bibliografia	89

Sommario

Tra gli effetti dell'ictus cerebrale dopo la fase di stabilizzazione si denota un'insorgenza di deficit motori. La riabilitazione ha il compito di contribuire alla riduzione della condizione di handicap connessa alla disabilità data dalla patologia, lavorando sui livelli di mobilità degli arti colpiti da paresi. Lo scopo dello studio trattato nella seguente tesi è quello di investigare gli effetti che possono portare diversi tipi di feedback audio nell'esecuzione di alcuni movimenti riabilitativi da parte dei pazienti. A tale scopo sono stati presi in considerazione 47 soggetti sani che non presentano alcun tipo di anomalia motoria legata all'arto superiore dominante. Quattro diverse tipologie di feedback audio sono state sviluppate via software, ed altrettanti erano i gruppi con i quali sono stati suddivisi i diversi soggetti. Ogni soggetto era sottoposto al medesimo esercizio di target tracking quindi ciascun gruppo differiva dall'altro non in base al tipo di esercizio che doveva svolgere perchè era il medesimo per tutti, bensì per la tipologia di feedback audio al quale i soggetti appartenenti a quel gruppo venivano sottoposti. Per ciascun soggetto è stata valutata l'accuratezza della sua performance mediante il calcolo di svariati indici di errore. La valutazione dell'effetto del feedback uditivo è stata ottenuta per mezzo di un confronto incrociato fra gli indici dei diversi gruppi ai quali sono stati applicati dei particolari test statistici.

Introduzione

Ridurre l'incidenza dell'ictus sull'indipendenza dei pazienti anziani durante la vita quotidiana sta diventando uno degli obiettivi primari della società nei paesi sviluppati, a causa dell'invecchiamento della popolazione e per l'incidenza di ictus nelle persone anziane [1, 2]. La riabilitazione ha un ruolo fondamentale nel ridurre le disabilità dei pazienti colpiti da ictus, sia durante il ricovero (fase acuta/subacuta, riabilitazione ospedaliera) sia dopo la dimissione (fase cronica, riabilitazione ambulatoriale/in casa). Dal 1995, quando fu presentato il primo pionieristico sistema robotizzato per la riabilitazione, è stato studiato l'uso della robotica nella riabilitazione post-ictus in modo approfondito, ottenendo dei risultati clinici positivi anche se non troppo soddisfacenti. Questo articolo presenta i più recenti sviluppi in questo campo di ricerca innovativo, delineando l'andamento degli studi e il ruolo futuro della robotica nella riabilitazione post-ictus.

L'uso di sistemi robotici e mecatronici nella riabilitazione post-ictus è un nuovo approccio terapeutico che, di recente, ha attirato l'attenzione di una crescente comunità di gruppi di ricerca sia di medicina che di ingegneria[3, 4]. Finora sono stati proposti molti protocolli terapeutici e soluzioni tecniche, questi si differenziano in diversi modi, nel tipo di robot e di cinematica, nel livello di interazione robot-paziente (feedback), nel tipo di esercizio, nella durata della pratica e in base alle caratteristiche del paziente [4]. Non tutti i sistemi proposti sono stati sottoposti a rigorosi test clinici, alcuni di loro sembrano essere stati progettati da un punto di vista solo ingegneristico piuttosto che per un fine terapeutico. Tuttavia, lo sforzo dedicato a questo settore da parte di molte istituzioni e società indica che il potenziale della robotica nella riabilitazione post-ictus è ancora poco

sfruttato [4, 5].

Il presente studio rientra in un progetto più grande, che il gruppo di robotica del DIMEG dell'Università di Padova porta avanti da alcuni anni [6] [7] [8] [9], il quale approfondisce il ruolo che il feedback audio gioca nel motor learning e cerca di comprendere quali vantaggi una stimolazione sonora porti nell'esecuzione di task motori, in vista di un suo utilizzo in un dispositivo di riabilitazione. In particolare, finora sono stati condotti esercizi di target tracking, durante l'esecuzione dei quali venivano forniti diversi tipi di feedback audio per capire quali effetti essi portassero nell'esecuzione di alcuni movimenti riabilitativi.

Proseguendo su questa strada, questa tesi si propone di investigare gli effetti che diversi tipi di feedback portano nell'esecuzione di alcuni esercizi di target tracking, confrontando in particolare la modalità video con quella audio. Per l'esecuzione dei test è stato utilizzato un joystick a due gradi di libertà, controllato da un software implementato in MatLab e Simulink.

Nel primo capitolo vengono presentati ed i risultati della ricerca bibliografica effettuata ed una revisione della letteratura, punto di partenza per l'ideazione e lo sviluppo di un nuovo tipo di test. Nel secondo capitolo viene introdotto il sistema sviluppato per l'implementazione dell'esercizio di tracking. Nel terzo capitolo viene descritta in dettaglio la procedura con cui sono stati effettuati i test. Nel quarto capitolo, infine, sono presentate le elaborazioni effettuate sui dati raccolte durante lo svolgimento delle prove e i principali risultati ottenuti.

In questo contesto, l'obiettivo è quello di capire quali tipi di feedback siano più adatti a fornire certe informazioni: in particolare, verranno confrontati feedback legati agli stessi parametri, ma forniti in due modalità differenti, quella audio e quella video. Ci si aspetta che le diverse modalità siano più adatte a rappresentare grandezze diverse.

Capitolo 1

Revisione della Letteratura

In questo capitolo vengono presentati i risultati della ricerca bibliografica: come motori di ricerca sono stati usati *PudMed*, *EngineeringVillage* e *Google*. La ricerca spazia tra articoli degli ultimi dieci anni e è stata ampliata a studi di anni precedenti grazie ai riferimenti bibliografici, per gli articoli di maggiore interesse. Sono state utilizzate le seguenti parole chiave:

- motor learning
- motor control
- feedback (audio/video)
- force field
- KR (knowledge of result)
- KP (knowledge of performance)

Alla fine del capitolo vengono presentate due tabelle che riassumono il contenuto della revisione della letteratura effettuata. La prima si riferisce ai soli articoli citati in questo contesto, la seconda è completa di tutti.

1.1 Il ruolo della robotica riabilitativa

Ridurre l'incidenza dell'ictus sull'indipendenza dei pazienti anziani durante la vita quotidiana sta diventando uno degli obiettivi principali della società nei paesi

sviluppati, a causa dell'invecchiamento della popolazione e dell'incidenza di ictus nelle persone anziane. La riabilitazione ha un ruolo fondamentale nel ridurre le disabilità dei pazienti colpiti da ictus, sia durante il ricovero (fase acuta/subacuta, riabilitazione ospedaliera) sia dopo la dimissione (fase cronica, riabilitazione ambulatoriale/in casa). Dal 1995, quando fu presentato il primo pionieristico sistema robotizzato per la riabilitazione, è stato studiato l'uso della robotica nella riabilitazione post-ictus in modo approfondito, ottenendo dei risultati clinici positivi anche se non troppo soddisfacenti. Questo articolo presenta i più recenti sviluppi in questo campo di ricerca innovativo, delineando l'andamento degli studi e il ruolo futuro della robotica nella riabilitazione post-ictus. L'uso di sistemi robotici e meccatronici nella riabilitazione post-ictus è un nuovo approccio terapeutico che, di recente, ha attirato l'attenzione di una crescente comunità di gruppi di ricerca sia di medicina che di ingegneria. Finora sono stati proposti molti protocolli terapeutici e soluzioni tecniche, che si differenziano in diversi modi, nel tipo di robot e di cinematica (alcuni esempi sono rappresentati in figura 1), nel livello di interazione robot-paziente (feedback), nel tipo di esercizio, nella durata della pratica e in base alle caratteristiche del paziente [2]. Non tutti i sistemi proposti sono stati sottoposti a rigorosi test clinici, alcuni di loro sembrano essere stati progettati da un punto di vista solo ingegneristico piuttosto che per un fine terapeutico. Tuttavia, lo sforzo dedicato a questo settore da parte di molte istituzioni e società indica che il potenziale della robotica nella riabilitazione post-ictus è ancora poco sfruttato [2]. Ci sono due ragioni principali per il rapido sviluppo di questo campo di ricerca. Prima di tutto, ci sono prove sempre più certe che il sistema motorio dopo essere stato colpito da ictus assume un comportamento plastico e quindi può essere trattato con l'esercizio motorio [10, 11]. In secondo luogo, il numero di persone che necessitano di riabilitazione dopo l'ictus è in rapida crescita a causa dell'invecchiamento della popolazione [12]. Per questo motivo, in un prossimo futuro, i costi dovuti all'incidenza di ictus metteranno sempre più pressione sui bilanci della sanità. Negli USA ogni anno si verificano circa 795.000 nuovi ictus, che comprendono sia nuovi episodi che persone già precedentemente colpite, e per il 2009 la stima dei costi diretti e indiretti dovuti all'ictus è di 68,9 miliardi di dollari, mentre il costo della vita media di un paziente colpito da ictus

ischemico è stimato attorno ai 140.048 dollari [13]. Il costo per i pazienti in fase acuta nei 2 anni seguenti al primo caso di ictus copre il 45,0% del costo di una vita intera, a lungo termine le cure ambulatoriali ammontano al 35,0%, mentre i costi per le cure in casa sono il 17,5% [14]. Percentuali simili sono riportate da un recente studio dell' Erlangen Stroke Registry in Germania [15]. In questo contesto, è in aumento la domanda di nuove tecnologie che permettano l'attuazione di programmi riabilitativi più efficaci e convenienti. Ad oggi, l'ictus è la prima causa di disabilità a lungo termine nei paesi sviluppati: tra il 15 e il 30% delle persone colpite rimangono disabili, il 20% richiedono cure mediche a 3 mesi dopo l'insorgenza [13, 16, 17]. L'obiettivo principale della riabilitazione per soggetti affetti da ictus (emiplegico) è quello di promuovere il recupero delle funzionalità perse, con lo scopo di riacquistare l'indipendenza nelle attività della vita quotidiana. Per garantire un esito funzionale positivo, i programmi di riabilitazione per ictus sono basati su esercizi ripetitivi task oriented [18]. Inoltre, la letteratura scientifica suggerisce che l'intervento di riabilitazione diventa significativamente più efficace quando viene garantita una stimolazione intensa, soprattutto se in fase subacuta (entro i primi 6 mesi) [12] e in una struttura specializzata [19]. Tuttavia, i programmi tradizionali di training dipendono molto dalla formazione del fisioterapista e dalla sua esperienza [20], e, talvolta, forniscono una quantità limitata di trattamenti per soggetto a causa della mancanza di terapisti [11], che è il problema crescente. Per di più, la valutazione della mobilità del paziente e delle sue capacità è fatta in gran parte da operatori sanitari basandosi su scale di valutazione standardizzate e approvate, ma pur sempre soggettive [21]. Questi fattori rendono difficile il confronto tra i diversi protocolli di trattamento.

Il potenziale dei sistemi robotici nella riabilitazione post-ictus è ampio e sfaccettato. I sistemi robotici sono adatti per produrre un intenso training motorio task-oriented per gli arti dei pazienti, sotto la supervisione di un terapeuta, come parte integrata di un insieme di strumenti per la riabilitazione, che dovrebbe poi includere anche approcci più semplici, non robotici [22]. In questo modo si può alleviare l'intensa fatica dovuta alla riabilitazione, permettendo al fisioterapista di concentrare gli sforzi sul recupero funzionale durante la terapia individuale e di supervisionare diversi pazienti allo stesso tempo durante le sessioni di terapia

robot-assistita [23]. Questo metodo permetterebbe di sfruttare meglio le competenze e il tempo dei fisioterapisti, aumentando, allo stesso tempo, l'efficacia e l'efficienza del programma di riabilitazione. Tuttavia, la riabilitazione robotica non è importante solo per il fatto di aumentare la qualità e l'intensità dell'esercizio. Infatti, i sistemi robotici possono essere utilizzati non solo per elaborare esercizi semplici e ripetitivi, ma anche per creare nel paziente una stimolazione multisensoriale più complessa e controllata (ad esempio stimolazione visiva, uditiva e di forza). In questo modo, può essere prodotto un più elevato livello di stimolazione rispetto a quella solitamente utilizzata durante la terapia hand-over-hand [2]. Un feedback estrinseco può essere, inoltre, utilizzato per dare al paziente la knowledge of results (KR) e/o la knowledge of performance (KP) durante il training robotico, facilitando così il raggiungimento degli obiettivi del movimento e stimolando il soggetto a impegnarsi nello svolgere l'esercizio di riabilitazione [24]. Un altro potenziale vantaggio dei sistemi robotici sta nel fatto che sono in grado di misurare una serie di parametri cinematici e dinamici durante il movimento dell'arto del paziente, permettendo di valutare diversi indicatori relativi alle prestazioni del paziente sia durante che dopo l'esercizio (per esempio la gamma del movimento, velocità, morbidezza, ecc.) [2]. Questi valori possono essere utilizzati per quantificare i progressi del paziente in maniera più obiettiva rispetto a una scala di valutazione clinica ma anche per adattare l'esercizio di riabilitazione in base alle esigenze specifiche del paziente. D'altra parte, i parametri ingegneristici proposti finora nella letteratura scientifica sono in genere legati allo specifico hardware robotico utilizzato e/o al tipo di esercizio attuato, e sono ben lungi dall'essere considerati come valida alternativa alle tradizionali scale di valutazione [21]. L'accettazione della tecnologia robotica[25] da parte dei pazienti e dei fisioterapisti può esser esso stesso un problema, anche se non risulta essere la principale preoccupazione per i dispositivi sviluppati fino ad oggi [26].

Per quanto riguarda il paziente, un punto chiave è quello di evitare che il soggetto si annoi usando uno strumento. Questo obiettivo può essere raggiunto implementando un esercizio che sia stimolante, quasi un gioco, con carico su misura del paziente, promuovendo un ruolo attivo di quest'ultimo nella riabilitazione [2]. D'altra parte, la riduzione del contatto fisico con una persona può

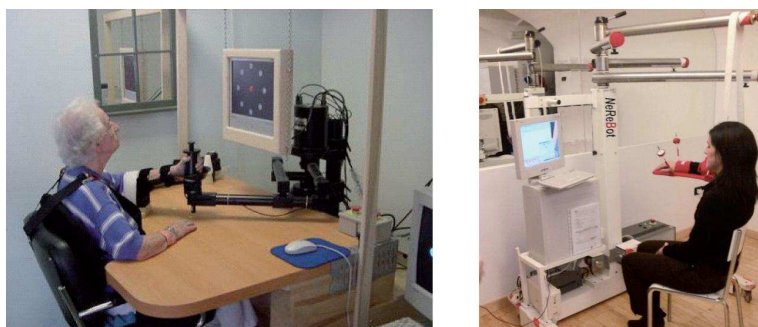


Figura 1.1: Due esempi di dispositivi robotici per la riabilitazione: MIT-Manus e NeReBot

diventare uno svantaggio [11], e può portare a considerare la robotica come una minaccia per i terapeuti, piuttosto che come un sostegno e un'alternativa in più per l'esercizio dei pazienti. Tuttavia, il divario culturale tra i fornitori di tecnologia, i professionisti di riabilitazione e gli utenti finali sta diminuendo gradualmente grazie alle conoscenze acquisite negli ultimi tempi.

Nonostante i potenziali benefici dei movimenti di training mediati da robot nel post-ictus [22], l'efficienza clinica di questo approccio è ancora in discussione. Anche se sono stati misurati dei benefici motori, un'importante critica riguarda la mancanza di informazioni che mostrino miglioramenti sulle misure ADL (Activity of Daily Living) rispetto alla terapia tradizionale [27, 28]. Questo è un risultato chiave per il futuro della riabilitazione post-ictus mediata da robot, così come un recupero funzionale è fondamentale per il reinserimento dei soggetti malati nella vita sociale e domestica, che rimane l'obiettivo principale dei programmi riabilitativi. Tuttavia, anche l'equivalenza tra terapia fisica e robotica può essere considerata come un risultato positivo, che consente l'introduzione dei sistemi robotici nella pratica clinica almeno per promuovere un efficace uso delle risorse umane [11] (come un terapeuta che può gestire numerosi pazienti allo stesso tempo) e che consente la standardizzazione dei programmi riabilitativi. Per quanto riguarda il training robotico dell'arto superiore, gli studi randomizzati e controllati (RCTs) condotti fino ad oggi hanno dimostrato che i pazienti che hanno ricevuto training robot-assistiti in combinazione con la fisioterapia hanno più probabilità di migliorare la funzionalità motoria e la forza dell'arto paralizzato rispetto ai pazienti che

ricevono solo trattamenti fisici; comunque, nelle ADLs[28] non vengono riportate sostanziali differenze. Tuttavia, un confronto tra studi della fase acuta/sub-acuta e studi della fase cronica rivelano miglioramenti significativamente differenti nelle ADLs tra i due sottogruppi, suggerendo che un precoce intervento robotico può essere più efficiente in termini di risultati funzionali rispetto alla fase cronica del training robotico. Questi risultati devono essere interpretati con cautela, dal momento che sono stati utilizzati differenti sistemi, esercizi e protocolli di trattamento negli studi inclusi in questo articolo. Lo stesso emerge dal più recente articolo sull'uso dei dispositivi robotici per il gait training [6], in cui si dimostra che i pazienti che ricevono assistenza robotica nella camminata in combinazione con la fisioterapia hanno più probabilità di ottenere una camminata indipendente rispetto ai pazienti che si allenano senza questi dispositivi. Comunque, i risultati di altri due recenti studi clinici suggeriscono che la terapia manuale può essere ancora più efficace rispetto al training robotico dell'arto inferiore, sia in fase sub-acuta che in fase cronica [28, 6]. Una possibile spiegazione di questa scoperta può essere una riduzione nel controllo volontario della postura durante il training della camminata robot-assistita a causa della ritenzione della coppia bacino-tronco con l'assistenza passiva e altalenante fornita dal sistema robotico impiegato negli studi [29]. Un'altra motivazione può essere che entrambi i trials coinvolgono pazienti già in grado di deambulare, e i sistemi robotici potrebbero essere più adatti per persone non ancora in grado di farlo. Gli scarsi risultati funzionali dei primi RCTs possono essere interpretati considerando che la maggior parte dei dispositivi robotici per la neuro-riabilitazione esistenti, sono designati e programmati per produrre semplici pattern di movimenti stereotipati degli arti dei pazienti (tipicamente, traiettorie lineari in uno spazio di lavoro planare o 3D), spesso non legati alle attività funzionali incluse nelle misure ADL. Inoltre, viene tipicamente usata una realtà virtuale invece di una realtà mista e/o un contesto di training ambientale, questo può ostacolare il trasferimento dei risultati dei training alle situazioni di vita quotidiana [2]. Infine, il tempo e lo sforzo spesi dai pazienti per imparare a interagire con l'ambiente robotico/virtuale può ridurre l'efficacia del training robot-assistito per quanto riguarda la manipolazione degli oggetti reali. Tuttavia, un confronto diretto tra approcci robotici e tradizionali basati

sulla maggioranza delle misure ADL può essere fuorviante. Infatti, durante la riabilitazione standard, ai pazienti vengono spesso insegnate delle strategie compensative che li aiutano a compiere il task utilizzando l'arto non paralizzato [30] e permettendo loro di acquisire valori ADL più alti anche in presenza di un modesto recupero dell'arto malato. Al contrario, le strategie di compensazione di solito non sono esercitate durante il training robotico. Per questa ragione, differenti e nuove valutazioni possono essere usate per quantificare correttamente l'efficienza dell'approccio robotico al livello funzionale. Grazie ai risultati positivi ottenuti fin



Figura 1.2: Il Pnew-WREX , un dispositivo robotico - esoscheletro - guidato da attuatori pneumatici.

qui, ai bisogni crescenti e all'attenzione data a questa tecnologia medica a causa dell'invecchiamento della popolazione, l'uso di sistemi robotici nella riabilitazione post-ictus crescerà ulteriormente nel prossimo futuro. La maggior parte delle limitazioni dei robot per la neuro-riabilitazione saranno in parte o totalmente superate adattando la tecnologia attuale e/o in fase di sviluppo, in accordo con i recenti progressi nel campo della riabilitazione verso approcci orientati alla funzionalità [2]. Inoltre, gli strumenti robotici saranno probabilmente integrati in nuovi sistemi in concomitanza con altre tecnologie emergenti (ad esempio, stimolazione elettrica e sistemi di training basati su sensori[11, 20]). Tuttavia, le esigenze di mercato e le pressioni delle imprese possono favorire la diffusione dei dispositivi

robotici esistenti, prima che una nuova generazione di robot riabilitativi più efficienti diventino commercialmente disponibili; questo permetterebbe di provare l'attuale tecnologia robotica su una scala più ampia e di facilitare la valutazione degli effettivi benefici economici legati all'introduzione dei sistemi robotici nella pratica riabilitativa, non solo a livello ospedaliero, ma anche considerando l'impatto del training robotico sul costo della vita dei pazienti colpiti da ictus. Questa valutazione può diventare un fattore chiave per rendere l'approccio robotico più attrattivo per le compagnie d'assicurazione e i sistemi di assistenza sanitaria, con l'obiettivo di creare un interesse commerciale a lungo termine nei sistemi di riabilitazione robotica. Inoltre, l'uso della tecnologia robotica sarà probabilmente estesa a differenti campi di riabilitazione, come la vasta gamma di malattie neurodegenerative. Per esempio, sono stati presentati recentemente i risultati di test clinici con dispositivi robotici sulla sclerosi multipla e su pazienti pediatrici con lesioni corticali[31]. Ricerche future chiariranno se, attraverso la progettazione tecnica e/o nuovi esercizi e protocolli di trattamento, i task ADL potranno essere realmente migliorati dal training robotico. Come suggerito dai risultati dei primi test clinici su dispositivi per l'arto superiore, un campo da esplorare è il trattamento robotico della fase acuta e subacuta , così da capire se un precoce e intensivo intervento robotico può migliorare significativamente il risultato funzionale, durante il ricovero ospedaliero del paziente [22, 31]. Da un punto di vista tecnico, una sfida importante sarà la creazione di dispositivi robotici in grado di allenare il paziente a compiere movimenti più naturali e funzionali[2, 31, 30]. La seconda sfida sarà quella di sviluppare nuovi sistemi robotici in grado di modulare correttamente il livello di assistenza, per evitare che il paziente riduca il proprio sforzo se non in presenza di affaticamento. Idealmente, i sistemi robotici dovrebbero essere adattati in modo da fornire il minimo livello di assistenza di cui necessita il paziente, mantenendo il rapporto di causalità tra lo sforzo del paziente e il movimento risultante dell'arto anche quando viene fornita assistenza robotica[4, 32]. Un altro miglioramento nei programmi di riabilitazione robotica ci si aspetta possa derivare dall'applicazione dei principi del motor learning computazionale alla neuro riabilitazione, per fornire un quadro di riferimento per la progettazione di protocolli di riabilitazione ottimali [33]. Tuttavia, la conoscenza

su come operi il motor learning durante la neuro-riabilitazione non è ancora sufficiente per imporre il design del dispositivo di terapia robotica [34], sebbene siano state proposte recentemente alcune indicazioni in questa direzione [35]. I progressi deriveranno probabilmente dalla futura ricerca sull'uso del feedback uditivo e multimodale nella riabilitazione assistita da robot, per aiutare a mantenere impegnato il paziente in tutta la sessione di terapia[36]. Infine, saranno necessari ulteriori test clinici per chiarire quali tipi di approcci progettuali (per esempio, esoscheletri vs sistemi robotici end-effector based, training bilaterale vs unilaterale, approccio prossimale vs il distale, training complesso vs il non-complesso e fase di trattamento acuto-subacuto vs la fase cronica) e quale parametri pratici possono essere più efficienti per specifici gruppi di pazienti. Il mercato sanitario globale sta crescendo più velocemente rispetto al prodotto interno lordo(PIL), dopo aver aumentato il proprio punteggio da 8 a 8.6% del PIL mondiale tra il 2000 e il 2005. In termini assoluti, al netto dall'inflazione, questo rappresenta una crescita del 35% nella spesa mondiale per la sanità in un periodo di 5 anni [37]. Quindi, la spinta ad usare efficientemente le risorse dell'assistenza sanitaria sta diventando sempre più elevata, e vengono proposti approcci differenti per migliorare il rapporto costo-efficacia dei programmi riabilitativi, includendo ospedali, case di cura e riabilitazione nell'ambiente domestico [38, 39]. Mentre al giorno d'oggi la robotica rappresenta un approccio realistico nei primi due ambienti, vi è ancora un bisogno insoddisfatto di dispositivi con prezzi accessibili per la riabilitazione in casa[22, 40]. La sicurezza è una delle preoccupazioni principali per tali sistemi dato che possono essere azionati senza supervisione, limitando così il ruolo del team di riabilitazione alla pianificazione e al monitoraggio remoto (sia online che offline) dell'intervento. In questo contesto, la realtà virtuale può essere un valido approccio alternativo, sebbene tali sistemi, da soli, non possono fornire assistenza ai pazienti; in questo modo essi sono adatti solo ad allenare parzialmente i soggetti. Una seconda, fondamentale questione dei dispositivi di riabilitazione da utilizzare in casa è il costo, dovuto al loro basso tasso di utilizzo rispetto ai sistemi di riabilitazione e di assistenza ambulatoriale e ospedaliera. Tuttavia, il numero di sopravvissuti all'ictus disposti ad acquistare un dispositivo di training casalingo a basso costo dovrebbe essere ragionevolmente elevato, soprattutto se si considera

che la durata dei programmi di riabilitazione ambulatoriali spesso non soddisfano le esigenze e le aspettative dei pazienti cronici [3]. In realtà, essi di solito offrono poche settimane di trattamento all'anno. Pertanto, un maggior sforzo di ricerca sui sistemi di riabilitazione casalinghi è previsto in un prossimo futuro ed i primi dispositivi a basso costo saranno probabilmente sviluppati e immessi sul mercato. La disponibilità di questa tecnologia consentirebbe la rivisitazione di tutto il programma di riabilitazione, che dovrebbe essere auspicabilmente basato sull'uso di dispositivi standard [22], per aiutare sia i pazienti e gli operatori sanitari a prendere confidenza con la tecnologia senza il bisogno di un riapprendimento delle nuove modalità di interazione in ogni fase del processo di recupero.

1.2 Apprendimento Motorio

L'apprendimento delle capacità motorie svolge un ruolo fondamentale nello svolgimento delle attività quotidiane[41]. È noto che le performance relative allo svolgimento di qualunque tipo di esercizio migliorano con la pratica. Il tasso di apprendimento e le prestazioni finali, avvengono non solo in base ad un elevato numero di nozioni aggiuntive che possono specificare nel dettaglio il tipo di task, ma anche grazie alla "qualità" della pratica, ovvero alle condizioni nelle quali il soggetto si trova a lavorare. Sempre riguardo al concetto di qualità, una potenziale via per migliorare l'apprendimento è legata all'applicazione di un feedback aggiuntivo nelle fasi di apprendimento che possa integrare i responsi ottenuti mediante vista, udito e propriocezione. Lo studio del *motor learning* viene affrontato da due punti di vista differenti riassumibili in KR[42] (knowledge of results) e KP[43] (knowledge of performance). Nel primo caso al termine del task vengono fornite al soggetto informazioni che mettono in relazione il risultato del movimento compiuto con l'obiettivo originale. Nel secondo caso invece durante il test vengono fornite al soggetto informazioni legate alla natura del movimento in questione.

1.2.1 Motor Control

Per *motor control* intendiamo la capacità del sistema nervoso di regolare o dirigere il movimento. Quest'ultimo è spesso descritto nel contesto dell'esecuzione di una particolare azione; quando si studia il controllo motorio, lo si fa in relazione ad azioni specifiche: cammino, corsa, raggiungimento (reaching), controllo della stazione eretta. In pratica si studia come viene controllato il movimento nel contesto di una specifica attività, assumendo che ciò fornisca informazioni sui principi generali. Il movimento, poi, non può essere studiato senza tenere conto della percezione. Per "percezione" si intende l'integrazione dei dati grezzi sensoriali in informazioni fruibili cognitivamente. È importante sottolineare che percezione e azione sono essenziali l'una all'altra: la percezione, cioè, non è un meccanismo passivo, ma attivo, che mira ad anticipare le conseguenze sensoriali di un'azione, legando quindi le informazioni sensoriali e i comandi motori in un 'tessuto computazionale' coerente. I sistemi sensoriali/percettivi forniscono informazioni sullo stato del corpo (ad esempio, la posizione del corpo nello spazio) e sulle caratteristiche dell'ambiente, critiche per la regolazione del movimento (ad esempio, la presenza di ostacoli). Un ultimo aspetto, non meno importante dei precedenti, è il legame del movimento con processi cognitivi quali l'attenzione, la motivazione, le emozioni; sono questi ultimi a determinare la formazione dell'intento o obiettivo del movimento.

1.2.2 Motor Learning

Il motor learning è un insieme di processi interni che portano con la pratica a cambiamenti relativamente permanenti nella capacità di movimento. L'acquisizione di capacità motorie viene introdotta in tre fasi: conoscitiva, associativa e autonoma. L'obiettivo dell'apprendimento è quello di automatizzare gradualmente le abilità attraverso queste fasi. Nella fase cognitiva il soggetto si concentra sul capire cosa deve fare, ed è una fase che prevede un gran numero di errori in termini di prestazioni proprio perchè il soggetto sta cercando i modi più svariati di risolvere il problema. È proprio in questa fase quindi che al soggetto viene proposto un feedback particolare che gli permette un'agevolazione dell'apprendi-

mento. Da un punto di vista ingegneristico, con l'obiettivo di stimolare il processo di apprendimento motorio del paziente, si dovrebbe sapere che tipo e la quantità di stimoli a cui viene sottoposto il paziente. Per esempio, alcuni risultati sperimentali si basano sulla riduzione degli errori cinematici dovuti ad adattamento. Per elaborare ciò è necessario utilizzare strategie di controllo che permettano all'utente di fare errori e nello stesso tempo di essere consapevole dei suoi errori. Di solito, questo tipo di informazioni è resa al paziente attraverso il feedback visivo, che viene utilizzato per riprodurre un compito virtuale come quello di far seguire ad un soggetto un target di riferimento che si muove all'interno di uno spazio virtuale[44]. Il feedback audio può essere impiegato per amplificare ed enfatizzare piccoli errori di cinematica inoltre il suono è particolarmente adatto per visualizzare le informazioni relative alla velocità, la cui derivazione dal feedback visivo sarebbe necessariamente molto più complessa da elaborare da parte del tester. L'integrazione di un feedback audio potrebbe anche essere una soluzione promettente per superare alcuni comuni problemi di visualizzazione (come ad esempio occlusione visiva). Riassumendo: la percezione visiva fornisce informazioni riguardo l'ambiente esterno e consente la previsione di eventi futuri, ma il tempo di elaborazione riferito alle informazioni uditive è nettamente più veloce rispetto ai tempi di reazione visiva, e ciò lo rende più importante per la reazione posturale agli stimoli disturbanti. Questi aspetti devono essere ulteriormente approfonditi, con l'obiettivo di comprendere se e come feedback uditivo può aumentare la stimolazione dei pazienti ed è proprio quello che si cercherà di fare in questo progetto di tesi.

1.2.3 Feedback sonoro

Alcune recenti ricerche in ambito neurologico hanno dimostrato che la stimolazione sonora è in grado di aumentare la plasticità cerebrale, influenzando specifici meccanismi neuronali che possono contribuire al recupero da un danno neurologico, come può essere l'ictus. Alcuni studi analoghi hanno dimostrato che un feedback sonoro ha prodotto netti miglioramenti in alcune attività di routine quotidiana (come la distribuzione del peso nel sedersi, la camminata) in pazienti affetti da malattie neurologiche (Parkinson, SM, Ictus). In seguito a queste ricerche in

campo medico, l'ingegneria riabilitativa ha introdotto nei nuovi dispositivi medici un feedback sonoro che può agevolare il paziente sia nell'apprendimento dell'esercizio sia mantenere alta la propria concentrazione su ciò che sta svolgendo. L'*auditory feedback*, propriamente detto, denota quindi un segnale audio, generato automaticamente dal dispositivo, in risposta ad un'azione dell'utente o in risposta ad uno stato interno del sistema. I feedback possono essere suddivisi in due categorie [42, 43]:

- Riguardo la conoscenza dei risultati KR[45]: danno un'indicazione di come è stato svolto un esercizio o se è stato raggiunto l'obiettivo dell'esercizio stesso;
- Riguardo la conoscenza della performance KP [46]: danno un'indicazione dei movimenti eseguiti dal paziente nel portare a termine un esercizio.

In merito alla definizione sopra citata, i feedback audio più utilizzati sono quelli che forniscono informazioni sugli errori di movimento del paziente, quindi appartengono alla seconda categoria. Essi hanno duplice funzione :

1. Facilitare l'apprendimento del percorso e del movimento previsto dall'esercizio.
2. Incrementare l'attenzione e la motivazione del paziente durante l'esercizio [47].

Un esempio molto significativo dell'utilizzo di questi feedback integrati in un robot-riabilitativo è dato dal Pneu-WREX (Figura 1.2). Al paziente, vincolato a questo esoscheletro, veniva chiesto di seguire un target su un display e, allo stesso tempo, di identificare un distruttore che compariva sullo schermo stesso. Come illustrato nell'esempio precedente, l'importanza di un efficace feedback audio acquista valore negli esercizi di riabilitazione robot-assistiti. Il paziente in questa tipologia di riabilitazione, infatti, non ha l'ausilio del fisioterapista e il feedback audio è l'unico sistema che incoraggia il paziente stesso a mantenere la propria attenzione sull'esercizio e, quindi, a migliorarsi sempre. Il lavoro di ricerca oggi si pone dunque come obiettivo quello di individuare quali siano i parametri più

opportuni ed efficaci a cui legare un feedback (acustico o visuale) in modo che esso possa essere veramente d'aiuto al paziente nell'esecuzione dell'esercizio.

1.3 Obiettivi del lavoro

L'obiettivo preposto per tale progetto è lo sviluppo di una particolare tipologia di esercizio di target tracking. Tale test prevede l'implementazione di due diverse tipologie di feedback audio da sottoporre on-line ai 47 soggetti sani reperiti per il test. I seguenti dovranno svolgere l'esercizio avendo a disposizione come dispositivo di ingresso un joystick a due gradi di libertà. Il test in questione denominato *Variable Amplitude* viene sottoposto a ciascuno dei soggetti ed è strutturato in tre fasi distinte. L'elevato numero di soggetti è risultato necessario per l'ipotesi di studio che ci si era preposti inizialmente, la quale consiste nel sottoporre i soggetti al medesimo tipo di esercizio, ma nel contempo fornire loro tipologie di feedback audio differenti. Ecco perchè è stata effettuata una suddivisione randomizzata dei soggetti in quattro gruppi. Per la realizzazione del test è stata messa a disposizione un interfaccia grafica argomento recente di tesi dei nostri colleghi Andrea Gazaneo e Marco Battiston. Tale interfaccia grafica è risultata un ottimo punto di partenza per lo sviluppo della tesi, ma nel contempo il riadattamento di questa per il test che si è voluto sviluppare non è stato così immediato.

1.3.1 Software sviluppato: Accenni

Mediante l'interfaccia grafica debitamente riadattata al tipo di test, l'utente era in grado di caricare un particolare esercizio riabilitativo da sottoporre ai soggetti. Tale esercizio è definito di target tracking ovvero il soggetto sotto analisi deve essere in grado di inseguire un target che comparirà sullo schermo mediante un joystick 2DOF. Il test è stato strutturato in tre fasi distinte ciascuna delle quali però prevedeva un movimento del target di riferimento lungo una traiettoria rettilinea piana. Sostanziale differenza fra le tre fasi appena accennate è legata alla scalatura del movimento che veniva presentata a video rispetto l'arco reale che il soggetto compiva con il joystick. La lettura del movimento compiuto dal soggetto con il joystick è stata resa possibile mediante un *RealTimeModel* realizzato

con Simulink il quale gestiva la lettura delle coordinate evidenziate dal joystick mediante la corretta configurazione di un Encoder (incaricato della lettura delle variazioni di movimento rispetto l'asse delle ascisse) e di un potenziometro (incaricato della lettura delle variazioni di movimento rispetto l'asse delle ordinate). La generazione del feedback audio è stata deputata al software Pure Data con il quale l'interfaccia grafica creata era in grado di comunicare, via UDP, mediante l'invio di pacchetti dati contenenti i parametri legati a posizione e velocità del target di riferimento e del target rappresentativo del soggetto. Mediante questi dati, sono state realizzate due tipologie diverse di feedback audio: una legata all'errore di posizione del paziente rispetto al target, un'altra legata alla velocità del target di riferimento. Il feedback audio legato alla variazione della velocità del target è stato applicato sempre alla velocità del target di riferimento, ma da punti di vista differenti, ovvero:

1. nel primo caso faceva riferimento alla velocità del target relativa al video
2. nel secondo caso faceva riferimento alla velocità del target relativa al joystick

1.3.2 Test sui soggetti

Dopo aver lavorato sulla parte prettamente di programmazione, ci si è dedicati per tempo a reperire un numero adeguato di soggetti da sottoporre al test. A ciascun soggetto, prima di sostenere il test, è stato chiesto di compilare un questionario nel quale venivano presentate le più svariate domande utili per l'esaminazione futura dei dati, perchè prendevano in considerazione diversi aspetti legati alle abilità motorie del soggetto nel maneggiare joystick o strumenti simili, e informazioni che riguardavano problemi legati a vista, udito o manomazioni fisiche che avrebbero potuto compromettere l'esito del test. È stato considerato un numero complessivo di 47 soggetti sani che, mediante randomizzazione, sono stati suddivisi in quattro diversi gruppi (uno da 11 soggetti e tre da 12) e ciascun gruppo è stato sottoposto a diverse tipologie di feedback audio:

- NF-NoFeedback
- APER-AudioPositionErrorRelated

- ATRV-AudioTaskRelated_scalaVideo
- ATRJ-AudioTaskRelated_scalaJoystick

Ciascun soggetto, anche se sottoposto a feedback audio differenti, doveva svolgere la sessa tipologia di test suddivisa in tre parti distinte che denominiamo per questioni pratiche:

1. Pre_Test a lunghezza fissa
2. Post_Test a lunghezza fissa
3. Var_Test a lunghezza variabile

Modalità di esecuzione dei test, tipi di analisi effettuate e risultati ottenuti, saranno presentati nei prossimi capitoli.

<i>Anno</i>	<i>Autore</i>	<i>Argomento</i>	<i>Device</i>	<i>Soggetti</i>	<i>Feedback</i>
2007	DiPietro	KP	Mit-Manus	117s	video
2008	Mussa-Ivaldi	motor control learning	RIC robot	25h	audio(
1997	Conditt	motor learning	Hopkins robot	14h	force field
2009	Robertson	KP	RIC robot	12h	audio
2007	Taylor	KR/KP	RIC robot	12h	audio, force field
2010	Shabbot	motor learning	Pen on a digitizing tablet	25h	video, audio

Tabella 1.1: Elenco degli articoli citati in questo capitolo (nel campo *Soggetti* s = stroke, h = healthy)

Capitolo 2

Device sviluppato

2.1 Architettura del sistema

Nella figura sottostante viene raffigurata la postazione di lavoro (Figura 2.1) dove sono stati eseguiti i test in questione. I soggetti venivano fatti sedere sulla sedia e veniva chiesto loro di indossare le cuffie li presenti.



Figura 2.1: Postazione di lavoro dove sono stati eseguiti i test

L'esercizio di target tracking sviluppato veniva raffigurato nel monitor presente in figura posto esattamente di fronte al soggetto che eseguiva il test. Per l'esecuzione dell'esercizio ciascun soggetto agiva su un joystick a due gradi di libertà posto in prossimità del soggetto. Il sistema di controllo del joystick è stato implementato mediante il software Simulink di Matlab, mentre una scheda

di acquisizione collegata al quadro elettrico e in comunicazione con il computer, si occupava di gestire l'alimentazione degli azionamenti e gestire le varie fasi dell'esercizio attraverso un'interfaccia GUI debitamente realizzata per l'obiettivo preposto. Tale interfaccia è stata realizzata in Matlab e funge da riferimento per le comunicazioni fra il modello Simulink sviluppato, il software Pure Data e i dispositivi di input (joystick) e output (monitor e cuffie) in questione. PureData è il software deputato alla generazione dei suoni utilizzati nel test trattato. In figura vengono evidenziati i collegamenti logici tra le varie parti del sistema sviluppato.

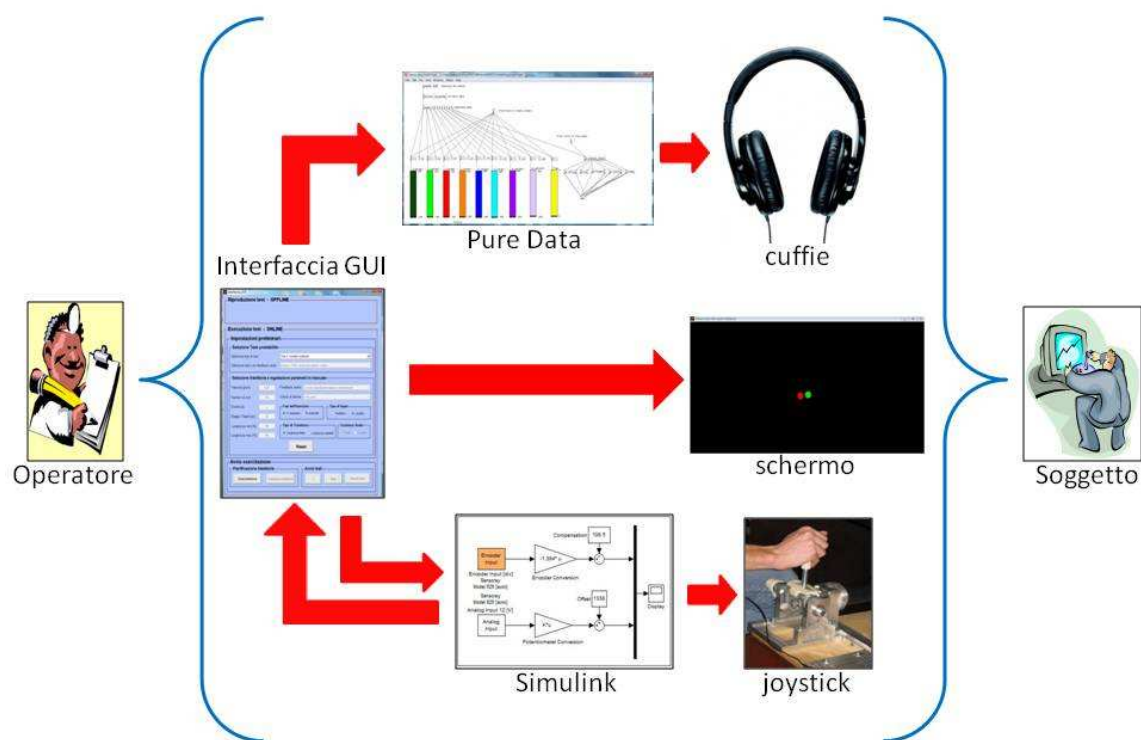


Figura 2.2: Schema logico e relative interconnessioni tra i software usati

2.2 Implementazione dell'interfaccia GUI

GUI è l'acronimo di graphical user interface e può essere definita come la “via” attraverso la quale un programma interagisce in maniera grafica con l'utente. Dal punto di vista informatico viene definito come un *paradigma* che mira a consentire all'utente di interagire con il computer manipolando graficamente degli oggetti, svincolandolo dall'obbligo di imparare una serie di comandi da impartire da tastiera sulla riga di comando. Nel seguente capitolo viene studiato nel dettaglio il programma sviluppato in MatLab per creare una tipologia specifica di esercizio definita Variable Amplitude.

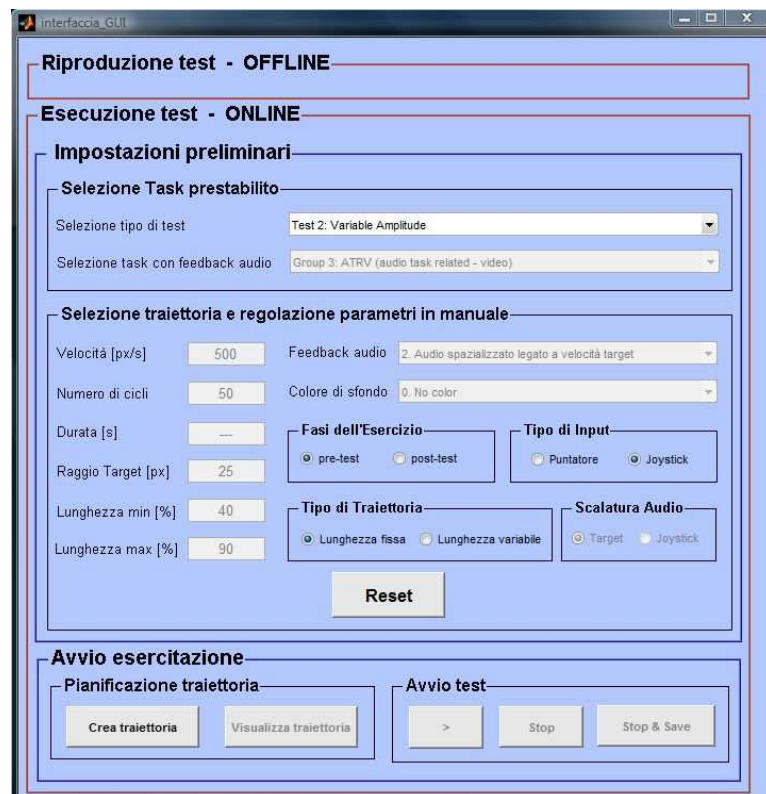


Figura 2.3: Interfaccia GUI realizzata

L'interfaccia GUI gestisce le varie fasi della creazione ed esecuzione del task permettendo la comunicazione tra Matlab, Simulink e PureData. In particolare, essa coordina le funzioni che creano ed impostano la traiettoria (*crea_tra*) e le funzioni che aggiornano la raffigurazione dei target durante l'esecuzione del test

(*draw*). La sincronizzazione e la ripetibilità delle varie fasi del test viene gestita dalla funzione *EventListener* contenuta nell'interfaccia GUI.

Creazione della traiettoria

Scelte le impostazioni dei parametri la creazione della traiettoria si articola in due fasi: la determinazione della coordinata di inizio e di fine di ogni singolo tratto e la pianificazione, secondo la legge desiderata, della traiettoria fra i punti iniziale e finale di ciascun tratto. La funzione che genera la traiettoria (chiamata *crea_tra*) viene avviata dal bottone apposito (Figura 2.12) visibile dalla GUI. Tale funzione richiede come ingressi alcuni dei parametri visibili dalla GUI, come la lunghezza minima, la lunghezza massima, il numero di cicli e la posizione iniziale da cui parte il target. La funzione *traj_pp* pianifica la traiettoria punto per punto, interpolando i valori assegnati da *build_trajectory2* funzione in grado di generare un vettore contenente in ciascuna cella la sola coordinata x che fa riferimento al cambio di direzione del target. I punti di cambio direzione ottenuti vengono raccordati tra loro mediante la seguente funzione interpolatrice polinomiale di quinto grado:

$$q(t) = a_0 + a_1t + a_2t^2 + a_3t^3 + a_4t^4 + a_5t^5 \quad (2.1)$$

Tale funzione genera una traiettoria la quale non presenta discontinuità in nessuno dei profili di interesse come posizione, velocità ed accelerazione. L'equazione prevede per l'istante iniziale e finale valori di velocità e di accelerazione nulli, così da ricavare un profilo di velocità simmetrico a campana con una caratteristica globale di dolcezza della traiettoria che è ben approssimata dal criterio di 'minimum-jerk'.

La funzione *crea_tra*, accessibile direttamente all'interno della GUI, sfrutta le funzioni *build_trajectory2* e *traj_pp* per costruire la traiettoria desiderata.

Tramite la funzione *crea_tra* viene salvata una struttura dati chiamata '*exercise_parameters*' che contiene tutti i parametri che identificano il task. Nello specifico il primo campo è la matrice '*exercise_parameters.data*', nella quale vengono salvate tutte le variabili relative al soggetto e al target, come le posizioni e le velocità in x e in y . Tale matrice viene aggiornata ogni qual volta l'*eventListener*

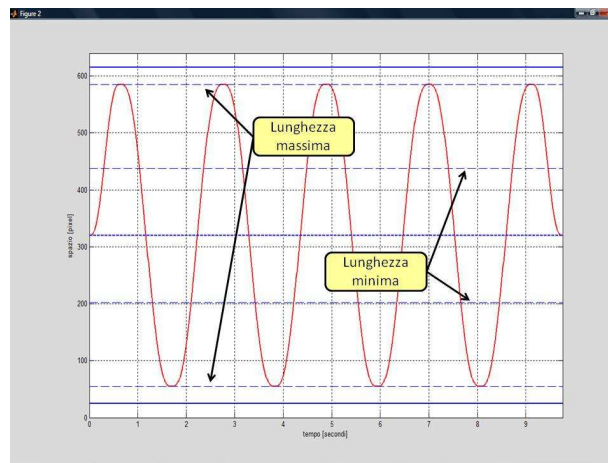


Figura 2.4: Esempio di traiettoria con 10 cicli

richiama la funzione *draw* durante l'esercizio. Altri parametri che vengono salvati sono:

- velocità del target
- numero di cicli
- lunghezza minima
- lunghezza massima
- raggio del target
- dimensioni dello schermo
- tipo di traiettoria selezionata
- stringa che identifica la traiettoria selezionata
- grado della funzione interpolatrice
- etichette che identificano le variabili di *exercise_parameters.data*
- tipo di test
- tipo di task
- istanti di inizio ciclo.

2.2.1 Fattore di scalatura

Per come sono state pensate a priori le tre differenti fasi del test *Variable Amplitude*, il soggetto è sottoposto a diversi cambi di scalatura legati al movimento che esegue con il joystick. L'interfaccia GUI, in base alla fase del test in cui si sta sottoponendo il soggetto, effettua una scalatura dei parametri cinematici legati al movimento effettuato con il joystick dal soggetto stesso. Per eseguire ciò si sono creati due fattori di scalatura:

- **fatt_t**: il quale considera il valore di scalatura applicato al target visualizzato a video

$$[1 \ 1 \ 1 \ 1]$$

- **fatt_j**: il quale considera il valore di scalatura applicato al movimento effettuato con il joystick e in base alla fase considerata assume valori differenti

$$[1 \ 1 \ 2 \ 2] \quad [2 \ 2 \ 1 \ 1]$$

il primo è il fattore di scalatura del movimento del joystick utilizzato nella fase di *Pre_Test* nella quale si prevedeva il passaggio da una scala 1:1 (scala reale) ad una scala 1:2 (scala del joystick 'raddoppia') effettuato nello specifico tra il 25°-26° ciclo. Il secondo invece è il fattore di scalatura del movimento del joystick nella fase di *Post_Test* nella quale si prevedeva il passaggio da una scala già 1:2 a quella 1:1 (si ritorna alla scala reale) effettuato nello specifico tra il 21°-22° ciclo. (Per una spiegazione più dettagliata sulle "fasi del test" si rimanda al capitolo 3).

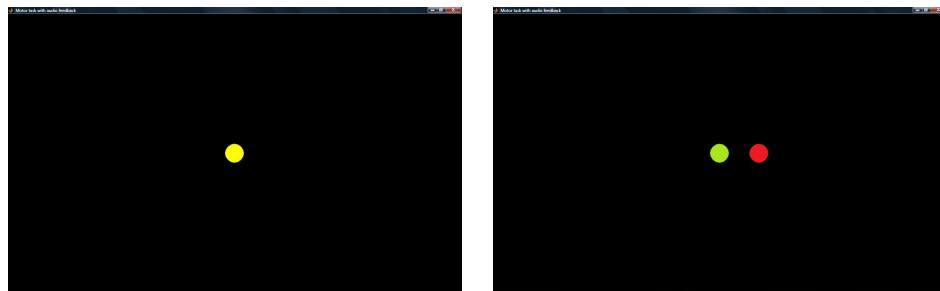
La scalatura nella fase di *Var_Test* non avvenendo in un ciclo esatto a causa della traiettoria variabile è stata realizzata in maniera diversa, o meglio:

1. è stata creata una matrice che prende in considerazione gli istanti di variazione della traiettoria
2. è stata eseguita un'interpolazione tra gli istanti in questione e il fattore di scala scelto

in questo modo ad ogni cambio di direzione il fattore di scalatura permette di agire di conseguenza.

2.2.2 Aggiornamento delle variabili: Funzione Draw

Soffermandosi alla raffigurazione dei target nel progetto precedente siamo stati spinti ad apportare delle modifiche concettuali: i target non saranno più considerati come “oggetti” bensì come “punti ingranditi”. L’inizializzazione di tali punti avviene nell’interfaccia GUI (Figura 2.5(a)) ma i continui aggiornamenti sulle coordinate di posizione dei target avvengono, scanditi dall’*EventListener*, nella funzione *draw* (Figura 2.5(b)).



(a) Inizializzazione dall’interfaccia GUI (b) Aggiornamenti grazie alla *draw*

Figura 2.5: Workspace

La funzione *draw* disegna soggetto e target in ogni istante temporale, salvando i dati aggiornati relativi all’esercizio nella matrice *exercise_parameters*. Tale funzione è responsabile inoltre della comunicazione, mediante la spedizione di pacchetti OSC, con il software dei suoni. La *draw* viene lanciata, alla frequenza di 100Hz, dalla funzione *eventListener* (implementata direttamente nell’interfaccia GUI) attraverso l’istruzione:

```
draw(obj,data,start_btn,ex,sim_data,sock,t)
```

i cui parametri in ingresso sono:

- *obj*: struttura che contiene i riferimenti dell’indicatore del soggetto e del target

- *data*: matrice che contiene i parametri che descrivono il task
- *start_btn*: riferimento al pulsante ‘Start’
- *ex*: struttura che contiene i parametri utili all’elaborazione finale dei dati
- *sim_data*: struttura che contiene, in ogni istante, le posizioni di soggetto e target
- *sock*: riferimento al socket UDP
- *t*: tempo di simulazione Simulink (corrisponde a *t_att*)

Dall’interfaccia GUI diversi parametri temporali (Figura 2.6) vengono passati alla funzione *draw*. Tali parametri temporali sono dei vettori i quali considerano:

- *t_att* = corrisponde al tempo dettato da Simulink (*t*) a cui vengono sottratti 10 secondi relativi all’intervallo che intercorre tra l’avvio del task e l’istante in cui inizia a muoversi il target, per cui all’avvio dell’esercizio *t_att* parte da -10s
- *t_sim* = definisce i tempi di simulazione nei quali il target effettivamente si muove

L’indice *indt* permette di scorrere *t_sim*: esso viene posto a 1 fino a che il target non inizia a muoversi, ovvero da quando *t_att* assume valori maggiori o uguali a zero, cosicché *indt* inizia a scandire *t_sim* con un periodo di campionamento pari a 0,01s, fino a quando il target non si ferma. Al termine dell’esercizio il vettore *t_sim* non viene più incrementato e *indt* assume un valore pari all’ultima posizione del vettore.

Prima della fase di esercizio vero e proprio vi è tempo sufficiente per permettere al soggetto di posizionarsi con il target che rappresenta i movimenti del joystick al centro dello schermo prima che il target inizi a muoversi. All’avvio dell’esercizio precede un countdown costituito da una successione di ‘beep’ della durata di 0.5 secondi intervallati da altrettante pause della stessa durata. Questa operazione è realizzata attraverso l’invio al PureData di pacchetti con tutti i

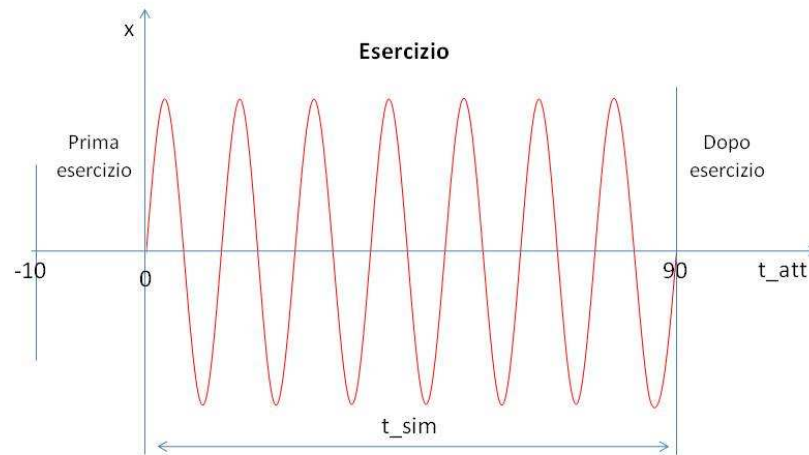


Figura 2.6: Relazioni tra i vari tempi utilizzati all'interno della draw

campi a zero tranne quello relativo all'audio, il quale è composto da una successione correttamente temporizzata di 0 e 9.

Nella fase di esercizio viene innanzitutto disegnato l'indicatore del target (pallino rosso). Poiché la velocità del soggetto non è disponibile a partire dalla GUI, viene calcolata attraverso la derivata numerica: in particolare, lo spazio percorso dal soggetto in un intervallo di tempo pari a 10 volte il periodo di campionamento (0,01s) viene diviso per l'intervallo di tempo stesso. Successivamente, solo quando il task lo prevede, viene impostato il colore dello schermo; a seconda della modalità di feedback video prevista, viene calcolato il parametro corrispondente:

- errore di posizione, pari alla distanza euclidea normalizzata rispetto a 6 volte il raggio del target
- errore di velocità, pari alla differenza di velocità tra soggetto e target, normalizzata rispetto a 1,5 volte la velocità massima del target

- velocità del target, normalizzata rispetto a 2 volte la la velocità massima del target

In seguito viene creato il pacchetto contente l'indice temporale, le posizioni di soggetto e target e il flag che identifica l'audio da spedire al PureData:

```
pd_packet = single([xy_patN(1) xy_patN(2) xytarN(1) xytarN(2)
    vtarN(1) vtarN(2) vpatN(1) vpatN(2) rem(audio_sel,100)]);
```

dove:

- xy_patN(1) e xy_patN(2): indicano rispettivamente la posizione normalizzata del soggetto rispetto all'asse x e y dello schermo
- xytarN(1) xytarN(2): indicano rispettivamente la posizione normalizzata del target rispetto all'asse x e y dello schermo
- vtarN(1) vtarN(2): indicano rispettivamente la velocità normalizzata del target rispetto alla direzione x e y dello schermo
- vpatN(1) vpatN(2): indicano rispettivamente la velocità normalizzata del soggetto rispetto alla direzione x e y dello schermo
- rem(audio_sel,100): rappresenta una funzione che permette di stabilire se il feedback audio percepito dal paziente fa riferimento al movimento del joystick (e quindi al movimento del paziente) oppure se deve essere scalato in base alle coordinate video (legato quindi al feedback visivo); in particolare, la funzione calcola un valore numerico per cui se risulta maggiore di 100 si ha la prima modalità, se minore la seconda. L'etichetta audio_sel specifica il tipo di audio in uso durante l'esercizio.

Al termine dell'esercizio viene implementato un unico beep che segnala il termine dell'esercizio. Infine, come ultima operazione, i dati aggiornati vengono salvati nella struttura *exercise_parameters.data*, attraverso l'istruzione:

```
exercise_parameters.data(ex.counter, :) = [t_att xy_pat(1) xy_pat(2)
    xtarV ytarV vtar_xV vtar_yV vpat_x vpat_y play_counter audio_sel
    tratto color_sel fatt_j(indt)];
```

dove:

- t_att : tempo della simulazione
- $xy_pat(1)$: coordinata x del soggetto
- $xy_pat(2)$: coordinata y del soggetto
- $xtarV$: coordinata x del target
- $ytarV$: coordinata y del target
- $vtar_xV$: velocità del target lungo x
- $vtar_yV$: velocità del target lungo y
- $vpat_x$: velocità del soggetto lungo x
- $vpat_y$: velocità del soggetto lungo y
- $play_counter$: numero di sospensioni e riavvi dell'esercizio
- $audio_sel$: flag che identifica il feedback audio selezionato
- $tratto$: numero del ciclo corrente
- $color_sel$: flag che identifica il feedback video selezionato
- $fatt_j(indt)$: fattore di scalatura

Il continuo aggiornamento della rappresentazione a display dei target, svolto come precedentemente spiegato dalla funzione *draw*, deve quindi prendere in considerazione che i parametri cinematici che caratterizzano la raffigurazione del target del soggetto devono essere debitamente scalati. Ecco le opportune manipolazioni effettuate su velocità del target del soggetto:

$$vtar_xV = vtar_x(indt)*fatt_t(indt)$$

$$vtar_yV = vtar_y(indt)*fatt_t(indt)$$

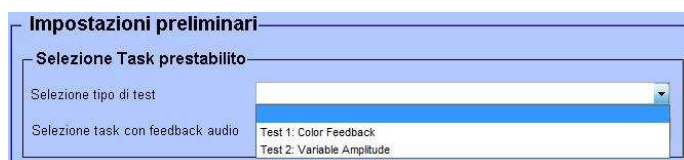
e sulla posizione del target del soggetto:

$$xtarV = xtar(indt)*fatt_t(indt)$$

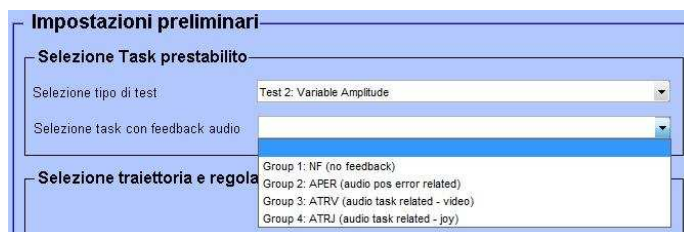
$$ytarV = ytar(indt)*fatt_t(indt)$$

2.3 Utilizzo dell'interfaccia GUI

L'interfaccia GUI sviluppata permette un facile approccio tra la macchina e l'operatore. Si è scelto di guidare l'operatore passo passo in ciascuna fase di selezione del task motorio. La schermata iniziale prevede la selezione del test a cui si vuol sottoporre il soggetto tramite un menù a tendina (Fig.2.7(a)). Selezionando il test 2: *Variable Amplitude*, ciascun soggetto reclutato sarà sottoposto al medesimo esercizio di target tracking ma da un menù a tendina è possibile scegliere quale feedback audio applicare al tester in esame (Fig.2.7(b)). Per gli obiettivi d'analisi pensati a priori, la selezione delle quattro diverse tipologie di feedback fin dall'inizio, crea una suddivisione dei soggetti nei quattro gruppi precedentemente accennati.



(a) Selezione test da sottoporre al soggetto



(b) Identificazione del gruppo a cui associare il soggetto

Figura 2.7: Prime fasi di caricamento del test mediante l'interfaccia GUI

Una volta impostato il test e il tipo di feedback audio applicato che identifica l'appartenenza ad un gruppo ben definito, sull'interfaccia vengono caricate automaticamente le impostazioni preliminari che caratterizzano la fase on-line del test. Tali impostazioni sono state fissate a priori dopo aver ripetutamente testato per primi l'esercizio di target tracking creato, e rese visibili sull'interfaccia in modo da permettere all'utente di avere ben chiaro con che parametri il soggetto in esame sta lavorando. Come si può notare dalla figura sottostante (Figura 2.3), numerosi sono i parametri che la caratterizzano. Nel dettaglio tali parametri

Figura 2.8: Parametri preimpostati una volta selezionato il gruppo di appartenenza

impostati rappresentano:

- Velocità del target fissata a 500 pixel/s per ciascuna fase del test
- Numero di cicli che variano in base alla fase del test considerata:
 - Pre_Test sono 50 cicli
 - Post_Test sono 40 cicli
 - Var_Test sono 30 cicli
- Raggio del target fissato a 25 pixel di diametro per ciascuna fase del test
- Lunghezza massima che corrisponde al percorso massimo che può compiere il target a video ed è fissato al 90% della lunghezza massima per ciascuna fase del test
- Lunghezza minima che varia in base alla fase del test considerato:
 - Pre_Test e Post_Test è pari allo 0% perchè comunque la lunghezza per le prime due fasi è costante e corrisponde alla lunghezza massima
 - Var_Test è pari al 40% ovvero il target non termina mai il ciclo in un punto esterno all'intervallo compreso fra lunghezza minima e lunghezza massima(vedi Fig.??).

- Feedback audio che presenta la descrizione per esteso della tipologia di feedback audio applicato e varia in base al gruppo selezionato nella fase preliminare
- Colore di sfondo che ai fini del nostro test rimane nero di default

Una volta considerati i valori dei parametri prefissati, l'utente può sottoporre il soggetto ad una tra le tre diverse fasi previste per l'esercizio sviluppato. Le fasi previste per tale esercizio sono tre e l'utente in base a come imposta i Radio-botton presenti nell'interfaccia imposta quale fase far sottoporre al test:

- Fase di Pre_Test (Figura 2.9(a))
- Fase di Post_Test (Figura 2.9(b))
- Fase di Var_Test (Figura 2.9(c))

(a) Impostazioni per fase di Pre_Test

(b) Impostazioni per fase di Post_Test

(c) Impostazioni per fase di Var_Test

Figura 2.9: Impostazioni delle diverse fasi del test *Variable Amplitude*

Oltre ai parametri considerati fin'ora, l'interfaccia GUI sviluppata permette all'utente di selezionare a cosa applicare la modalità di feedback audio impostato precedentemente [??] e con quale dispositivo di ingresso lavorare[2.3]. In entrambi i casi la selezione avviene mediante Radio-botton, dove:

- Nel primo caso le possibilità di dispositivo di ingresso con il quale far lavorare i soggetti sono:

1. joystick a due gradi di libertà ??

2. puntatore rappresentato dal mouse o dalla tavoletta grafica (usata nei test precedenti 3.1)



Figura 2.10: Selezione del dispositivo di input

- Nel secondo caso gli “oggetti” possibili a cui applicare la scalatura del feedback audio saranno:
 1. il target che compare a video
 2. il movimento effettivo relativo al joystick

Ma per l'esecuzione del nostro test è stata impostata la scalatura legata al target visualizzato sul monitor.



Figura 2.11: Selezione che permette di distinguere a “cosa” applicare il feedback audio in questione

Nell'eventualità che i parametri caricati e/o le fasi impostate precedentemente siano state caricate erroneamente, mediante il tasto di “Reset” è possibile reimpostare i parametri in questione.

Ad ogni modo una volta caricati i parametri è possibile creare [2.12(a)] e visualizzare [2.12(b)] la traiettoria che il target di riferimento effettuerà a video (Figura 2.12). I parametri sono stati impostati, la traiettoria è stata creata e visualizzata, arrivati a questo punto è possibile avviare il test mediante la finestra di avvio (Figura [??]): Al termine dell'esecuzione è possibile salvare i dati relativi all'esercizio svolto mediante il tasto “Stop & Save”, il quale permette un



Figura 2.12: Pianificazione Traiettoria



Figura 2.13: Finestra di avvio e di salvataggio

salvataggio nella cartella di riferimento. Le impostazioni di salvataggio sono di default:

T2-SiglaGruppo (NomeGruppoEsteso)-FaseEsercizio-...

alle quali volta per volta bastava inserire “Cognome” e “Nome” del soggetto testato.

2.4 Joystick a due gradi di libertà

Si tratta di una struttura a 2 gradi di libertà, realizzata presso i laboratori del DIMEG; essa comprende un grado di libertà attivo, comandato da un motore elettrico in corrente continua, e un grado di libertà passivo, comandato da un freno elettromagnetico on/off accoppiato con un giunto elastico. La leva del joystick è fissata direttamente all’asse del motore DC tramite un accoppiamento di superfici e un perno che consente alla leva di muoversi nella direzione dell’asse del motore. Sulla leva è inserito un anello di acciaio temprato che scorre appoggiandosi su 2 barre metalliche fissate su un supporto rotante che le mantiene perpendicolari all’asse di rotazione del motore; in questo modo si possono disaccoppiare i 2 g.d.l. associati all’azionamento attivo e a quello passivo. Il movimento di beccheggio della leva è limitato dai fincorsa, mentre il movimento di rollio è limitato dai bordi della cava realizzata sul supporto. Il range di movimento per entrambi gli assi è limitato a circa $\pm 30^\circ$.

La leva inizialmente era di lunghezza pari a 450mm ma, per agevolare i soggetti nell'impugnarla è stata realizzata una prolunga della leva in plastica. Da ciò risulta che:

- la *lunghezza* totale della leva era pari a $l = 160mm$,
- l'*arco di circonferenza*(spazio percorso dalla leva) è pari a:

$$\left(\frac{\pi}{3}\right) l = 167,55mm$$

- la *corda* da esso sottesa risulta:

$$2l \sin\left(\frac{\pi}{6}\right) = l = 160mm$$

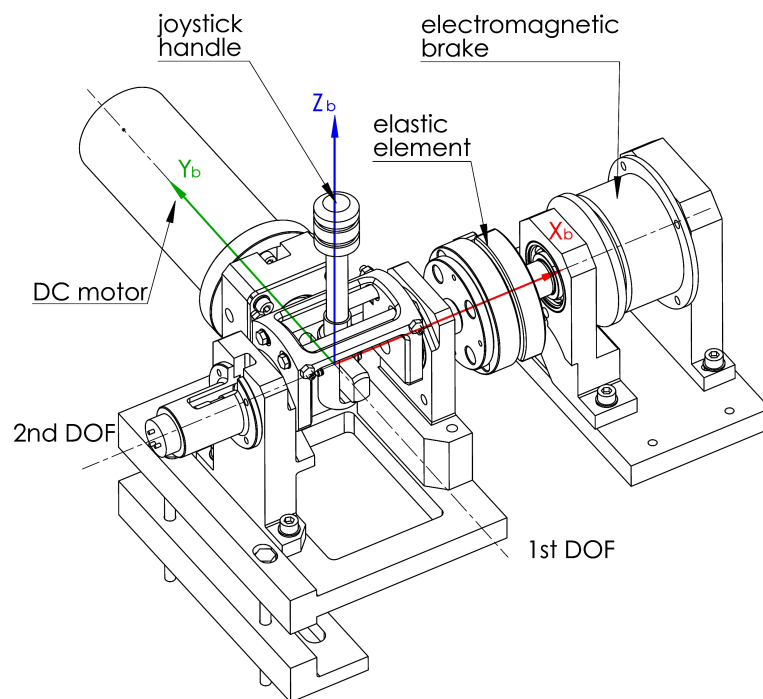


Figura 2.14: Immagine CAD del joystick utilizzato durante i test

Durante l'esecuzione dei test il device è stato utilizzato passivamente, ovvero il motore è stato tenuto spento ed il freno è stato rimosso in modo che il soggetto non potesse incontrare alcuna resistenza nel piano di lavoro e che gli fosse garantita la più completa libertà di movimento.

Il controllo del joystick, tramite l'ambiente di sviluppo Simulink, è operato da una scheda multifunzione I/O Sensoray modello 626, la quale permette di acquisire le informazioni fornite dai trasduttori (encoder e potenziometro) del joystick. La misura della rotazione della leva lungo l'asse di rollio è operata da un encoder ottico incrementale, acquisito in quadratura a 8000 step/giro, garantendo una risoluzione pari a $360^\circ/8000 = 0,045^\circ$. L'angolo di inclinazione della leva sull'asse di beccheggio, invece, è misurato tramite un potenziometro rotativo monogiro di precisione, realizzato in plastica conduttiva da $10k\Omega$ e alimentato a 10V. La conversione dei dati avviene mediante un ADC a 16 bit, impostato su un range di $\pm 10V$ (in questo modo vengono utilizzati solo metà dei valori disponibili: $2^{16} \text{valori} / 2 = 2^{15} = 32768 \text{valori}$), la risoluzione è pari a $360^\circ/32768 = 0,011^\circ$.

2.4.1 Simulink

Il controllo del joystick è effettuato attraverso l'implementazione di un modello in ambiente Simulink con Real-Time Windows Target, in modo da poter eseguire una simulazione in real-time. La figura 2.4.1 mostra lo schema del modello del

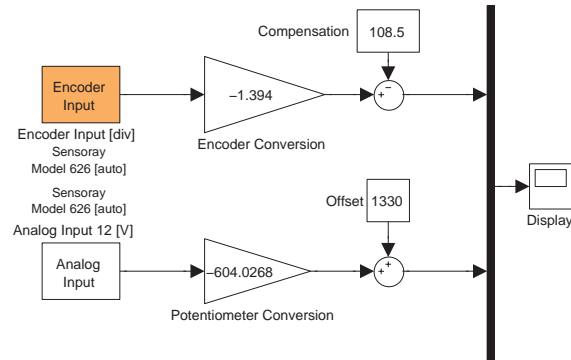


Figura 2.15: Modello Simulink realizzato

joystick implementato all'interno del programma Simulink: si notino, a sinistra, i blocchetti *Encoder Input* ed *Analog Input* relativi alla scheda Sensoray per l'acquisizione della posizione della leva del joystick rispettivamente da un encoder (asse x) e da un potenziometro (asse y). Tali blocchi sono direttamente disponibili nella libreria di Real-Time Windows Target e per la configurazione è sufficiente impostare i parametri ad essi connessi come l'indice del canale della scheda utilizzato e il tempo di campionamento che è stato fissato a 0,01s. Il gain e l'offset sono stati aggiunti, per ciascun asse, allo schema Simulink per passare dal sistema di riferimento nel workspace della leva del joystick a quello nell'area di lavoro definita dalla risoluzione (in pixel) di uno schermo Full HD (risoluzione 1920x1080). L'origine della terna di riferimento sullo schermo e quella di azzeramento dell'encoder all'avvio sono poste in basso a sinistra (con x positivo verso destra e y verso l'alto), posizionandosi frontalmente al sistema usato per i test sperimentali. Per ciascuno dei 2 blocchetti si tratta, quindi, di risolvere un sistema lineare a due equazioni in 2 incognite del tipo:

$$o_0 = ki_0 + v \quad (2.2)$$

$$o_1 = ki_1 + v \quad (2.3)$$

dove k e v rappresentano rispettivamente il guadagno e l'offset incogniti, i è il segnale in ingresso proveniente dall'encoder o dal potenziometro e o è l'uscita che rappresenta lo spostamento della leva, in pixel, misurato lungo l'asse x o l'asse y dello schermo.

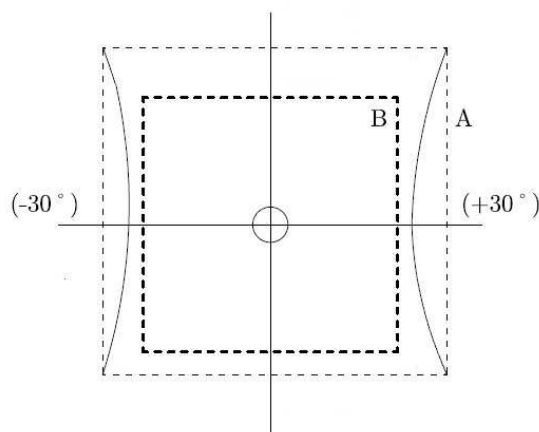


Figura 2.16: Area di lavoro della leva del joystick con evidenziati i limiti individuati tramite la procedura di calibrazione (A = limiti massimi dell'area di lavoro, B = area di lavoro effettiva)

È importante notare che lo spostamento della leva lungo i limiti della corsa utile genera un grafico riportato in figura 2.16, invece di un tracciato rettangolare: questa discordanza è dovuta alla cinematica del giunto del joystick. Per evitare di riprodurre il percorso dovuto alla cinematica sullo schermo, si è scelto di far corrispondere la lunghezza totale dello schermo alla distanza minima in ascissa dello spazio di lavoro del joystick. Ciò significa che per la trasformazione degli assi si è adottata una terza terna di riferimento spostata di un offset positivo nella direzione x rispetto alla terna di azzeramento dell'encoder in modo tale da compensare la cinematica. Di conseguenza il calcolo delle due incognite è stato effettuato leggendo direttamente il sensore di rollio (o di beccheggio) in posizioni note, cioè nell'origine e agli estremi dello schermo lungo l'asse x (o lungo l'asse y), e calcolando tramite le equazioni 2.2 e 2.3 i valori di k e v . Per quanto riguarda il segnale proveniente dall'encoder si ottiene $k = -1,394$ e $v = -108,5px$ mentre per il segnale proveniente dal potenziometro si ha $k = 604,0268$

e $v = -1330px$. Le posizioni del soggetto lungo i due assi, in ogni istante, vengono misurate dai sensori del joystick e convertite in Simulink. La funzione che da MatLab preleva il valore di Simulink, interrogando un “ascoltatore di eventi”, è chiamata *eventListener*, ed è direttamente implementata nell’interfaccia GUI. Le istruzioni relative alla memorizzazione dei dati sono riportate nelle seguenti righe di codice:

```
handles.sim_data.pat.x = block.InputPort(1).data(1) - offset(3)/2;
handles.sim_data.pat.y = block.InputPort(1).data(2) - offset(4)/2;
```

Il tempo della simulazione è generato dall’istruzione:

```
sTime = block.CurrentTime - handles.data.initial_pause;
```

Le posizioni e l’istante temporale vengono acquisiti dall’EventListener, la quale lancia la chiamata per la funzione che farà l’aggiornamento della grafica e dei feedback:

```
handles.exercise_data = draw(handles.obj, handles.data,
handles.start, handles.exercise_data, handles.sim_data,
handles.mysock, sTime);
```

La funzione *eventListener* viene eseguita automaticamente dal modello Simulink ad ogni ciclo di simulazione grazie alle impostazioni generate dalla funzione *loadModel*:

```
function mdl = loadModel(modelName);
mdl.modelName = modelName;
load_system(mdl.modelName);
mdl.viewing=struct('blockName','','blockHandle',[],
'blockEvent','','blockFcn',[]);
mdl.viewing.blockName = sprintf('s/Display',mdl.modelName);
mdl.viewing.blockEvent = 'PostOutputs';
mdl.viewing.blockFcn = @eventListener;
mdl.originalStartFcn = get_parm(mdl.modelName,'StartFcn');
```

e dalla funzione *addEventListener*:

```
handles.mdl.eventHandle{i} =  
add_exec_event_listener(handles.mdl.viewing(i).blockName,  
handles.mdl.viewing(i).blockEvent,  
handles.mdl.viewing(i).blockFcn);
```

2.5 Feedback audio

Per la generazione del feedback audio viene utilizzato il software freeware Pure-Data che sfrutta il protocollo OSC per ricevere in ingresso i dati relativi al suono da rilevare.

2.5.1 Regole di creazione dei pacchetti - protocollo OSC

L'Open Sound Control (OSC) è un protocollo di trasmissione che permette a strumenti musicali, computer ed altri dispositivi multimedia di scambiare 'music performance data' in tempo reale attraverso una semplice rete interna o Internet. L'OSC rappresenta un'evoluzione del MIDI (Musical Instrument Digital Interface), ovvero il protocollo standard, utilizzato fin dagli anni '80, per l'interazione degli strumenti musicali elettronici. La pressante domanda di riduzione dei costi, maggiore affidabilità e maggiore accessibilità a tutti gli utenti ha portato l'OSC a svilupparsi a tal punto da eguagliare e poi superare l'uso del MIDI.

Il protocollo OSC lavora ad altissima velocità sulla rete (broadband network speed) e permette di scambiare un maggior numero di dati in confronto al MIDI, dando inoltre maggiore flessibilità al tipo di dati a disposizione; tutto questo garantisce un livello di interazione più elevato tra macchine e software. I vantaggi nei confronti del MIDI sono molteplici: oltre alla differenza di velocità e di mole di dati che può essere trasmessa, i miglioramenti riguardano anche la risoluzione dei tipi di dato (il MIDI utilizza una risoluzione a 8 bit mentre con l'OSC si va ben oltre). La tecnologia alla base delle comunicazioni è la stessa utilizzata per lo scambio di dati su internet e permette di inviare diversi tipi di dati. Il protocollo di trasporto utilizzato per l'invio dei dati è l'UDP (user datagrams protocol), più leggero e veloce rispetto al TCP.

Tipi di dati fondamentali del protocollo OSC

- Int32: rappresenta un valore integer con segno a 32 bit in complemento a 2 memorizzati in formato big endian
- OSC-timetag: numero a virgola fissa a 64 bit memorizzati in formato big endian
- Float32: numeri a virgola mobile a 32 bit rappresentabili in formato IEEE754 memorizzati secondo il formato big endian
- Stringa: sequenza di caratteri composta da byte nel caso delle stringhe con codifica ASCII alla cui fine è posto un byte di 0 per rendere il numero totale di bit multiplo di 32 e quindi il numero di byte multiplo di 4.

OSC packets

Gli OSC packets (più comunemente noti come pacchetti) sono l'unità fondamentale di trasmissione del protocollo OSC. Tali pacchetti sono costituiti sia dai dati in essi contenuti scritti in formato binario, sia dalla dimensione di questi dati, tale dimensione deve essere sempre un numero di byte multiplo di 4. Inoltre, nell'atto di trasportare i pacchetti la rete deve tenere conto sia delle informazioni relative al contenuto sia di quelle relative alla dimensione, che devono essere entrambe trasmesse. I dati che vengono trasmessi in un pacchetto, riferito al protocollo OSC, si dividono in 2 tipologie: messaggio (message) o fascio (bundle). Il primo byte di dati permette di definire se il contenuto del pacchetto è del primo o del secondo tipo.

OSC Messages: è costituito da un OSC Address Pattern, ovvero una stringa che comincia con il carattere “\”, seguito da un OSC Type Tag String e da eventuali altri OSC Arguments. Un OSC Type Tag String è una stringa che comincia con il carattere “,”, seguito da una serie di caratteri che rappresentano il tipo di dato OSC Arguments a cui corrispondono. Qualsiasi altra stringa contenente caratteri non riconosciuti dal protocollo OSC viene automaticamente scartata

dal sistema. Una sequenza di OSC Arguments viene identificata dalla serie di rappresentazioni in codice binario di ogni singolo argomento.

OSC Bundles: (o fascio) è formato dalla stringa “#bundle” seguita da una Time Tag cui seguono zero o più Bundle Elements. La Time Tag è una timetag a 64 bit a virgola fissa. Un Bundle Element consiste della sua dimensione e del suo contenuto: la dimensione viene indicata da un int32, multiplo di 4, che rappresenta il numero di byte del contenuto, mentre il contenuto pu’o essere un Message o un altro Bundle.

2.5.2 Pacchetti creati dall’interfaccia

Dopo aver brevemente illustrato come deve essere espressa la sintassi dei protocolli OSC per l’invio dei dati ad un’applicazione esterna, si pu’o passare all’analisi dei pacchetti che l’interfaccia invia al software esterno (Pure Data) per la generazione del tipo di audio desiderato. Prima di tutto è necessario definire il dato da inviare, che nel caso specifico è composto da un vettore di dimensioni 1x9:

```
pd_packet = single([xy_patN(1) xy_patN(2) xytarN(1) xytarN(2)
vtarN(1) vtarN(2) vpatN(1) vpatN(2) rem(audio_sel,100)]);
```

dove:

- `xy_patN(1)` e `xy_patN(2)`: indicano rispettivamente la posizione normalizzata del soggetto rispetto all’asse x e y dello schermo
- `xytarN(1)` e `xytarN(2)`: indicano rispettivamente la posizione normalizzata del target rispetto all’asse x e y dello schermo
- `vtarN(1)` e `vtarN(2)`: indicano rispettivamente la velocità normalizzata del target rispetto alla direzione x e y dello schermo
- `vpatN(1)` e `vpatN(2)`: indicano rispettivamente la velocità normalizzata del soggetto rispetto alla direzione x e y dello schermo
- `rem(audio_sel,100)`: rappresenta una funzione che permette di stabilire se il feedback audio percepito dal paziente fa riferimento al movimento del

joystick (e quindi al movimento del paziente) oppure se deve essere scalato in base alle coordinate video (legato quindi al feedback visivo); in particolare, la funzione calcola un valore numerico per cui se risulta maggiore di 100 si ha la prima modalità, se minore la seconda.

L'etichetta `audio_sel` specifica il tipo di audio in uso durante l'esercizio.

La funzione `single`, applicata davanti a tutto il corpo da spedire, rappresenta la conversione necessaria per ottenere numeri a singola precisione. Tali numeri hanno una codifica a 32 bit che rispetta la dichiarazione di tipo `float` a 32 bit dell'OSC Message. Successivamente si utilizza il metodo:

per scomporre il vettore `pd_packet` in un vettore di `unit8`. Tale vettore è la rappresentazione in virgola mobile del dato trasmesso in modalità, ovvero per primo viene passato il byte più significativo. Un dato di tipo `single` (formato da 4 byte) viene scomposto in un vettore di 4 `unit8`, i quali rappresentano i suoi byte. Il valore `datastring` rappresenta il vettore. Lo stesso metodo viene utilizzato per predisporre i byte di 0 necessari per rendere sempre, in qualsiasi caso, il numero totale dei byte multiplo di 4. La stringa da inviare a Pure Data è la seguente:

```
string = char(['/pd_packet' zerostring(1:2) ',ffffffff'  
             zerostring(1:2) datastring]);
```

dove le 9 "f" indicano i 9 float che rappresentano i valori contenuti nel `pd_packet`.

2.5.3 Invio dei pacchetti

Per la comunicazione via UDP tra l'interfaccia realizzata e il software per la generazione dei suoni (PureData) sono stati utilizzati i metodi messi a disposizione dalla `pnet` di MatLab. La `pNet` è una particolare funzione input/output realizzata per la comunicazione via TCP/UDP in Matlab. Questa funzione può essere chiamata con diverse opzioni/comandi ed è in grado di supportare il trasferimento di dati di diverse tipologie e ordini di byte.

Il passo preliminare per la spedizione dei pacchetti creati consiste nella definizione

di un buffer di scrittura e di una connessione fra host e server. Innanzitutto si usa il comando:

```
mysock = pnet('udpsocket', 3455);
```

Questa istruzione crea il socket UDP e lo lega ad una specifica porta UDP. Su questo socket si possono ricevere i pacchetti UDP destinati alla porta specificata, e inviare pacchetti con questo socket come indirizzo sorgente. La funzione restituisce un valore maggiore di zero in caso di creazione andata a buon fine o -1 in caso di errore. Nel socket creato si possono destinare pacchetti UDP da inviare tramite il comando “`writepacket`”. Successivamente si crea la connessione fra la porta dell’host di destinazione ed il socket UDP, per evitare di specificare di volta in volta nel comando “`writepacket`” la porta di destinazione dei pacchetti:

```
pnet(mysock, 'udpconnect', '127.0.0.1', 3456);
```

In questo caso il valore 3456 identifica la porta remota che riceve il socket. Dopo aver ultimato il dato da inviare in accordo con le regole del protocollo OSC, si procede a scrivere il dato nel buffer del socket e ad inviarlo. I comandi a cui bisogna ricorrere sono:

```
pnet(sock, 'write', string);
```

per copiare i dati da inviare nel buffer di scrittura, e:

```
pnet(sock, 'writepacket');
```

che invia il contenuto del socket alla porta di destinazione, precedentemente specificata, sottoforma di pacchetto UDP.

2.5.4 Software di generazione di suoni: Pure Data

Pure Data è una piattaforma per la sintesi e l’elaborazione di audio, video e processi grafici in tempo reale. Ideato e sviluppato da Miller S. Puckette negli anni ‘90, Pure Data è comunemente usato per creare spettacoli di musica dal vivo, effetti sonori, composizioni musicali, analisi audio, interfacciamento con sensori, controllo di robot e anche per interagire con siti web.

L'unità funzionale di Pure Data è una scatola ('box'). I programmi vengono realizzati collegando tra loro una serie di scatole fino a formare un vero diagramma in cui si ha un flusso di dati continuo. Lo stesso programma è sempre in esecuzione, non c'è alcun tipo di separazione tra la scrittura del programma e l'esecuzione del programma stesso. Le potenzialità di questo open source sono illimitate, basta pensare che la comunità di utenti e programmatori di Pure Data hanno creato funzioni aggiuntive (chiamate 'esterne' o 'librerie esterne') che sono utilizzate per un'ampia varietà di utilizzi, come l'elaborazione video, la riproduzione e lo streaming di brani MP3 o video Quicktime, la manipolazione e la visualizzazione di oggetti 3D e la modellazione virtuale di oggetti fisici.

Nell'ambito dello studio presentato, si è fatto ricorso al software Pure Data per la generazione dei feedback audio e dei segnali sonori utilizzati nelle esecuzioni dei vari task. Si può dire che in questo frangente le applicazioni di Pure Data hanno svolto una funzione di server tra l'interfaccia per la gestione dei test e l'audio fornito nell'esecuzione degli stessi.

Capitolo 3

Test Sperimentali

Come inizialmente accennato il nostro lavoro nasce come proseguo di un progetto di tesi precedentemente trattata che riguardava sempre la realizzazione di esercizi di target tracking. Anche nel progetto di tesi cui si fa ora riferimento i soggetti erano sottoposti a diverse tipologie di feedback audio durante l'esecuzione dei test, e in base agli esiti hanno cercato di calutare se diverse tipologie di feedback audio portano a dei miglioramenti dal punto di vista delle performance.

3.1 Test Precedenti

Ciascuna sessione di test precedentemente trattata consisteva nella realizzazione di 6 tipologie di task differenti frutto della combinazione di tre feedback audio diversi (no audio, audio spazializzato relativo all'errore di posizione e audio spazializzato relativo all'errore di velocità) applicati rispettivamente a due tipi di traiettorie distinte (traiettoria con lunghezza fissa per ciascun ciclo e traiettoria con lunghezza variabile per ciascun ciclo). I task prevedevano una serie di movimenti(28) nella sola direzione orizzontale considerando che nel primo e nell'ultimo tratto l'estensione del movimento era dimezzata visto che il target partiva dal centro del Workspace e terminava nel centro del Workspace. Per i task a lunghezza fissa l'estensione dei tratti era al più pari al 60% della lunghezza massima, mentre per la traiettoria a lunghezza variabile l'estensione dei tratti oscillava tra il 20% e il 90% della lunghezza massima. Ciascun task aveva una durata pari a 80s e

la velocità media impostata per il moto del target era di 400pixel/secondo. Differentemente dal nostro progetto di tesi, il loro dispositivo di ingresso utilizzato per i test era una tavoletta grafica 3.1. Questa disposta orizzontalmente su un piano permetteva l'acquisizione dei movimenti dei soggetti i quali tracciavano i loro inseguimenti con un puntatore a forma di stilo sulla medesima.



Figura 3.1: Test precedente

In base ai test svolti in precedenza le principali conclusioni tratte prevedevano:

1. Qualsiasi sia il tipo di feedback uditivo utilizzato nell'esercizio, il task con traiettoria dai tratti variabili comporta una minore accuratezza nell'inseguimento rispetto al task con traiettoria dai tratti fissi.
2. L'audio sulla velocità del target rispetto all'audio assente ha l'effetto di ridurre il ritardo medio, fatto dovuto alla contemporanea riduzione del lag error e aumento del lead error. Il feedback uditivo sulla velocità del target ha come principale effetto quello di traslare nella direzione di anticipo il range di variabilità dei movimenti di correzione del soggetto sulla posizione del target durante l'inseguimento.
3. L'audio legato all'errore di posizione non dà significativi risultati rispetto alle performance in condizioni di assenza di audio

4. L'audio sulla velocità del target rispetto a quello sull'errore di posizione ha sulle performance gli stessi effetti riscontrati nel confronto con audio nullo: riduzione del ritardo medio e sensibile riduzione del lag comportano una riduzione nella ampiezza del range di variabilità degli errori nel task a traiettoria variabile.

3.2 Descrizione del test Variable Amplitude

Per la sessione di test in questione sono state suddivise le persone a disposizione (vedi cap.3.4.2) in quattro gruppi distinti. Qualunque soggetto che partecipava all'esercizio, indipendentemente dal gruppo di appartenenza, veniva sottoposto a tre fasi distinte del test definite per praticità:

- Fase 1 = Pre_Test
- Fase 2 = Post_Test
- Fase 3 = Var_Test

L'appartenenza a gruppi distinti è in funzione del tipo di feedback audio a cui i soggetti sono stati sottoposti ecco perchè i gruppi sono stati numerati:

- NF-NoFeedback
- APER-AudioPositionErrorRelated
- ATRV-AudioTaskRelated_scalaVideo
- ATRJ-AudioTaskRelated_scalaJoystick

Prima dell'inizio del test vero e proprio si è scelto di considerare una cosiddetta "Fase 0" nella quale veniva chiesto al soggetto di impugnare il joystick (il cui riferimento a video era un target verde), e di centrarsi sul corrispettivo target di riferimento (rosso) che si trovava al centro del display. Questa fase è risultata fondamentale ai fini pratici dell'esercizio, perchè veniva data la possibilità al soggetto di prendere dimestichezza con il joystick valutandone la sensibilità agli spostamenti, la velocità di risposta a video, la fluidità nella rappresentazione ed

altre proprietà simili, per permettergli di intraprendere il test vero e proprio con almeno un minimo bagaglio di conoscenza dello strumento di lavoro.

3.2.1 Fase di Pre_Test

Tale fase del test è caratterizzata da una totalità di 50 cicli compiuti dal target di riferimento a velocità pari a 500 pixel/secondo. Per i primi 24 cicli la scalatura a video del target del soggetto riferita al corrispettivo movimento del joystick, è una scala reale 1:1 (Figura 3.2(a)). Per l'intero 25° ciclo si ha una scalatura progressiva, tale che all'inizio del 26° ciclo la scala del soggetto diviene 1:2, ovvero raddoppia. Tale scalatura rimane fissa a 1:2 per i successivi 25 cicli (Figura 3.2(b)), ovvero fino alla fine della fase di Pre_test.

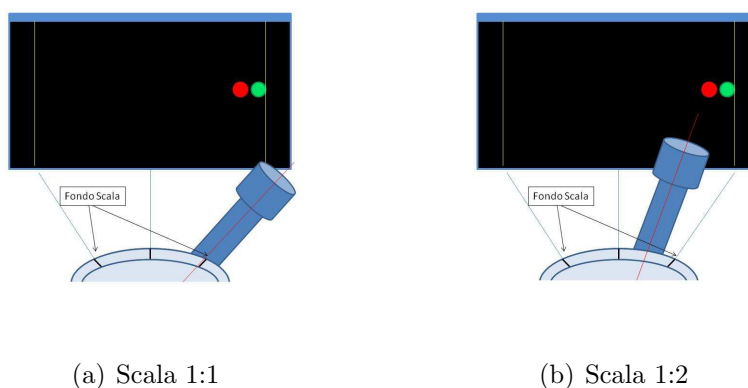


Figura 3.2: Variazione scalatura al 25° ciclo nella fase di Pre_Test

In tale fase cosa avviene: il soggetto per seguire il target di riferimento a cavallo tra il 25-esimo e il 26-esimo ciclo deve necessariamente correggersi o comunque modificare la propria tecnica di inseguimento, perchè si ritrova a lavorare con una scala del joystick diversa da quella con il quale ha iniziato a svolgere il test, nonostante a video l'ampiezza compiuta resti la medesima.

Tra la fase di Pre_Test e quella di Post_Test è stata inserita una pausa di 5 minuti nella quale al soggetto era permesso rilassarsi ed eventualmente togliersi le cuffie nel caso gli dessero fastidio. La pausa in questione di 5 min corrisponde ad un tempo che si trova spesso in letteratura, ecco perchè abbiamo scelto tale durata. Durante tale pausa venivano chiusi sia Matlab che Pure Data (vedi cap.3.5).

3.2.2 Fase di Post_Test

Nella seconda fase del test è stata prevista una totalità di 40 cicli compiuti dal target di riferimento a velocità pari a 500 pixel/secondo. Per i primi 20 cicli la scalatura a video del target del soggetto riferita al corrispettivo movimento del joystick, è quella lasciata alla fine della fase di Pre_Test, ovvero scala 1:2 (Figura 3.3(a)). Per l'intero 21° ciclo si ha una scalatura progressiva, tale che all'inizio del 22° ciclo la scala del soggetto ritorna ad essere in scala reale 1:1. La scalatura reale rimane fissa per i successivi 19 cicli (Figura 3.3(b)), ovvero fino alla fine della fase di Post_test. In tale fase cosa avviene: il soggetto per seguire il target

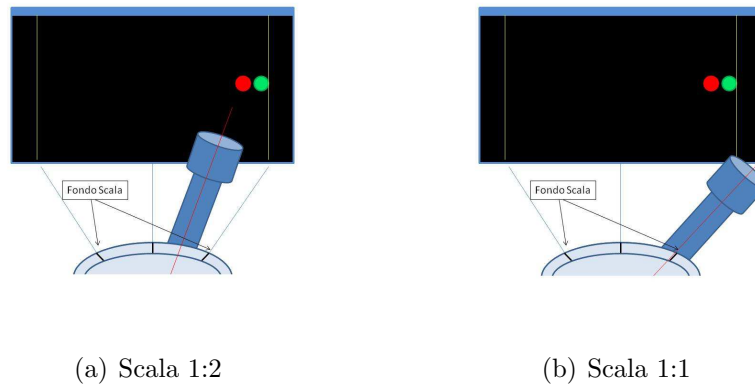


Figura 3.3: Variazione scalatura al °21 ciclo nella fase di Post_Test

di riferimento a cavallo tra il 20-esimo e il 21-esimo ciclo deve necessariamente correggersi o comunque modificare la propria tecnica di inseguimento, perchè si ritrova a lavorare con una scala del joystick diversa da quella con il quale ha iniziato a svolgere il test, nonostante a video l'ampiezza compiuta resti la medesima. Tra la fase di Post_Test e quella di Var_Test l'unica pausa concessa è stata quella necessaria per la chiusura di Matlab e Pure Data (vedi cap.3.5), poi il soggetto veniva sottoposto subito alla terza e ultima fase del test.

3.2.3 Fase di Var_Test

Nell'ultima fase del test sono stati previsti una totalità di 30 cicli. L'ampiezza di questi ultimi era soggetta ad una variazione continua e randomizzata ma pur sempre compresa fra gli intervalli impostati dall'utente mediante compilazione dei campi *Lunghezza fissa* ma soprattutto *Lunghezza variabile*.

Nell'immagine sottostante (Figura 3.4)abbiamo un grafico che rappresenta l'effettiva traiettoria visualizzata a video del target di riferimento nella fase di traiettoria a lunghezza variabile.

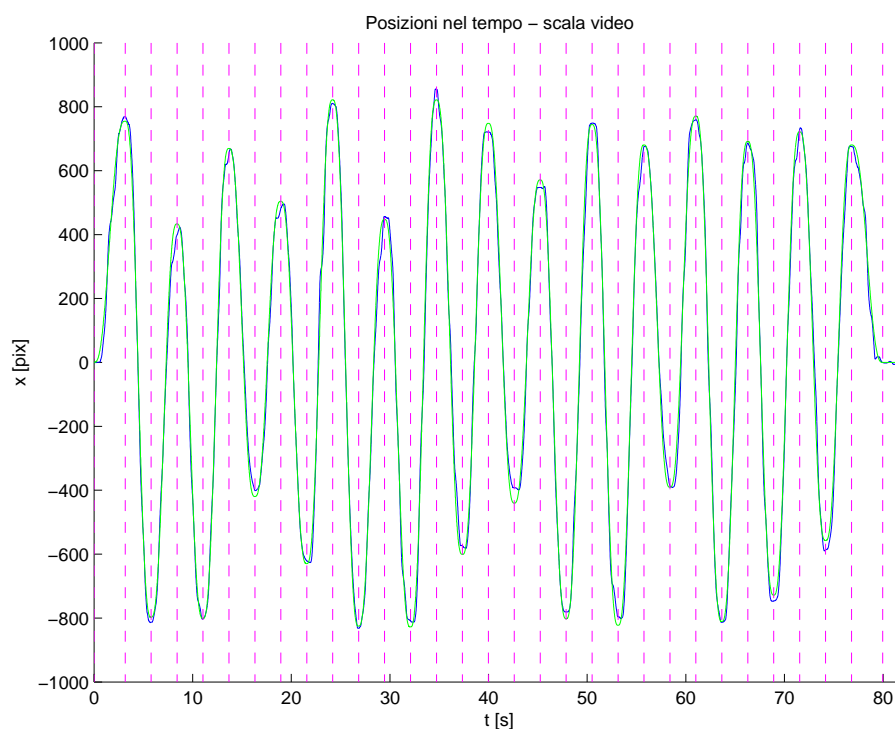


Figura 3.4: Rappresentazione traiettoria a video del target nella fase Var_Test

Istante per istante la scala per rappresentare il moto sullo schermo è stata fatta variare in maniera tale che alla fine di ogni movimento l'estensione necessaria da compiere con il joystick fosse sempre la stessa, ovvero l'estensione massima. La scalatura applicata al joystick risulta compresa tra 1:1 e 1:0,4.

A livello di codice si è quindi impostato che la scalatura istante per istante

risultava pari a:

$$1 : \frac{\text{AmpiezzaMovimentoTarget}}{\text{AmpiezzaMassima}} = \begin{cases} 1 : 1, & \text{agli estremi} \\ 1 : N \quad (N \text{ variabile}) & \text{fra gli estremi} \end{cases} \quad (3.1)$$

Nei grafici sottostanti (Figura 3.5) viene messa a confronto la traiettoria del target visualizzata a video (sovrapposta a quella effettivamente compiuta dal soggetto) con il grafico della progressiva scalatura del joystick.

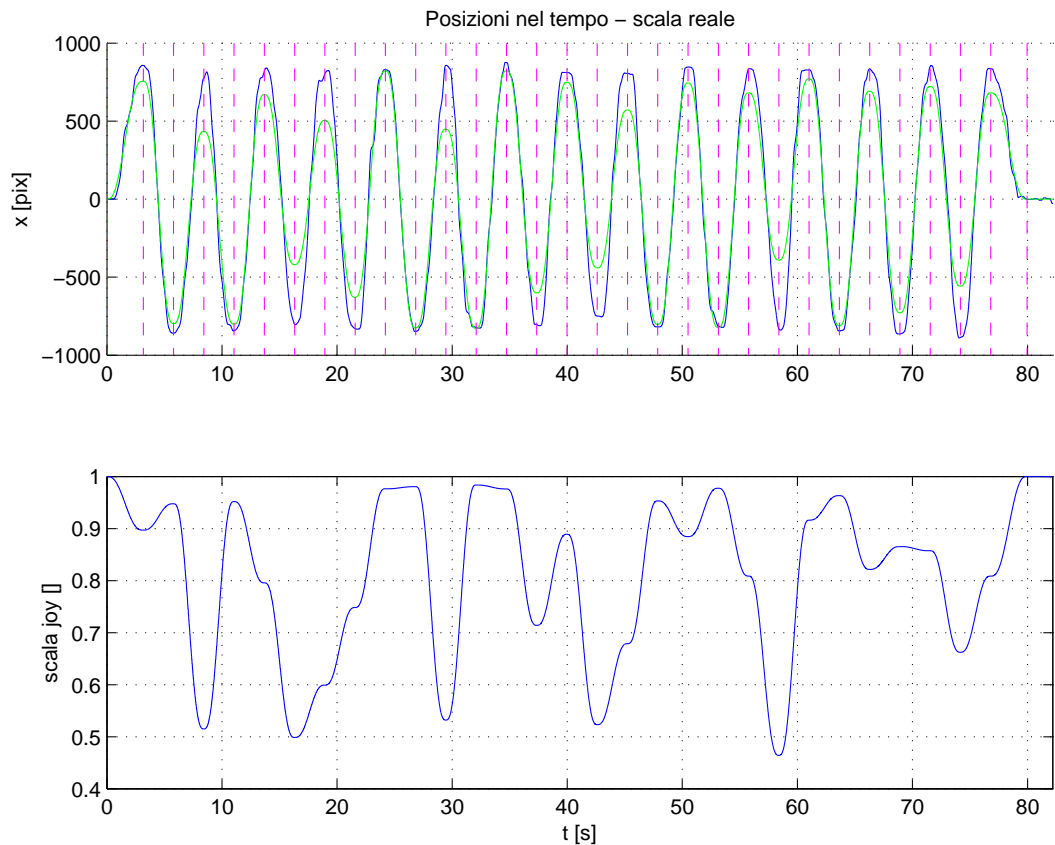


Figura 3.5: Traiettoria a video sovrapposta a traiettoria joystick, evidenziando scalatura di quest'ultimo

3.3 Tipologie di feedback audio applicati

Sebbene molte evidenze sperimentali supportano la tesi che l'utilizzo di un feedback in modalità audio incrementa il coinvolgimento in applicazioni che riguardano il completamento di task, è altrettanto vero che un feedback audio malamente progettato può risultare controproducente. Ad esempio se il suono è monotono o poco stimolante, o se le variazioni sonore non sono correlate a ciò che realmente sta accadendo nella scena virtuale, può accadere che i soggetti rinuncino all'utilizzo del feedback in quanto percepito solo come un elemento disturbatore. Sono necessarie quindi alcune linee guida in fase di progettazione del feedback che riguardano:

- il problema di evitare di creare un effetto di disturbo sul soggetto, anzi fare in modo che il feedback audio fornisca informazioni realmente utili
- la qualità del suono, che deve essere ottima
- la necessità di ridurre al minimo il tempo di ritardo con cui il feedback uditivo risponde ad un'azione del soggetto

Sulla base di queste considerazioni sono state considerate due diverse tipologie di feedback audio:

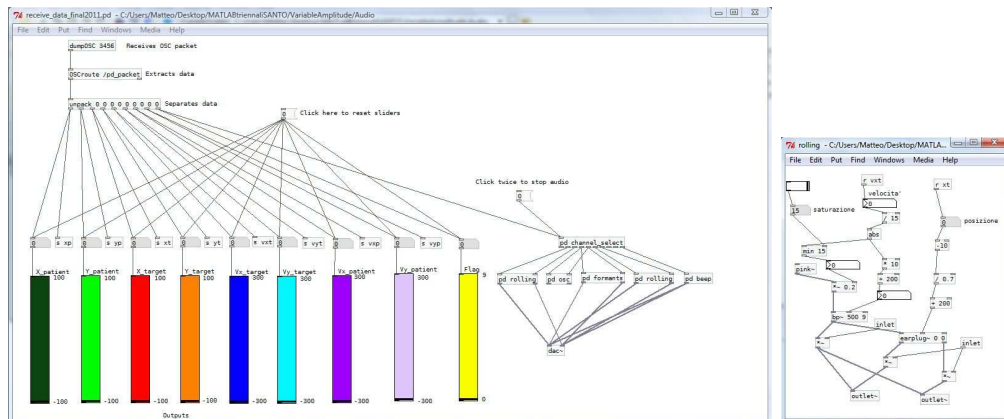
1. Audio spazializzato legato alla velocità del target
2. Audio spazializzato legato all'errore di posizione

Il primo tipo di audio riproduce un suono che simula quello del rotolamento di un oggetto su un piano, e l'effetto generale ottenuto è piuttosto soddisfacente in quanto il suono si integra perfettamente col movimento, creando proprio la sensazione che l'indicatore circolare del target (il pallino rosso) stia rotolando da una parte all'altra dello schermo durante l'esecuzione dell'esercizio.

3.3.1 Modifiche apportate ai suoni scelti

Essendo tale progetto un proseguo di una tesi trattata precedentemente, a partire dalle due tipologie di audio già implementate, si è cercato di apportare alcune

modifiche per cercare di rendere la tipologia di audio più adeguata possibile alle nostre esigenze.



(a) Finestra di lavoro principale di Pure Data

(b) Finestra di Talk Box

Figura 3.6: Interfaccia di Pure Data sfruttata per testare il funzionamento della GUI

Le modifiche apportate sono le seguenti:

1. *Prima Modifica:* Normalizzazione per i dati relativi alla posizione portati in percentuale con segno (± 100).
Il range di valori impostato per i dati relativi alla velocità venivano calcolati in base ai valori delle posizioni normalizzate e si estendevano in un campo compreso tra ± 300 , dove gli estremi sono stati scelti come i valori massimi corrispondenti ad un utilizzo non consono del joystick. Tali modifiche hanno comportato una correzione sulla spazializzazione del suono in cuffia.
2. *Seconda Modifica:* Per quanto riguarda il suono legato alla velocità del target, all'aumentare dell'errore prima vi era una sola variazione di intensità del suono, ora si è scelto di collegare la velocità con un guadagno ridotto legato oltre che alla intensità anche alla tonalità, per cercare di dare quell'effetto di "suono in lontananza" (esempio tipico la moto che sfreccia ad alta velocità).
3. *Terza Modifica:* Per quanto riguarda il suono legato all'errore di posizione rispetto al target di riferimento abbiamo fatto in modo di:

- eliminare la correlazione tra movimento lungo l'asse delle ordinate e il suono legato all'errore. In tal modo si è scelto di far dipendere la tonalità e la variazione di vocale del suono creato alla sola creazione di errore lungo l'asse x.
- abbiamo apportato una correzione al suono in modo da generare il cambio vocale ad errori in x molto grandi, mentre per errori ridotti si manteneva praticamente il suono di un'unica vocale.

3.4 Soggetti reclutati per il test

Per la tipologia di test considerato si è scelto di reperire un numero sostanzioso di soggetti in modo tale da avere una variegata gamma di persone a disposizione. A ciascuno di loro, prima di sostenere il test, è stato presentato un questionario 3.4.1 al quale è stato chiesto di rispondere. In aggiunta son stati fatti firmare ai soggetti due fogli distinti: il primo certificava la presenza del soggetto, il secondo consisteva in un foglio di "presa visione" il quale attestava che il test da svolgere veniva dettagliatamente spiegato prima di sostenere l'esercizio.

Il test elaborato era di una durata effettiva pari a circa 18 minuti compresa la pausa tra le fasi di Pre_Test e Post_Test, ecco perchè è stata stilata una tabella per la convocazione dei soggetti in modo tale da poterne testare uno ogni mezz'ora. I test sono iniziati il 17 Gennaio 2011 e si son conclusi il 22 Gennaio 2011, e son state testate rispettivamente 6 persone il 17/01, 15 persone il 18/01, 16 persone il 19/01, 8 persone il 20/01 e 2 persone il 22/01.

I test si sono svolti presso il *DIMEG(Dipartimento di Innovazione Meccanica e Gestionale)* sito in Via Venezia 1, Padova. È stata messa a disposizione una stanza nella quale si è cercato di limitare al minimo tutte le possibili fonti di distrazione per i soggetti. È stata regolata l'illuminazione sul display per evitare i riflessi della luce del sole, sono stati staccati tutti i telefoni presenti all'interno e durante l'esecuzione dei test non era permesso a nessuno entrare nella stanza.

Nelle diverse fasi dell'esercizio i soggetti sono stati minuziosamente monitorati per cercare di carpire se grossolani errori nel target tracking fossero condizionati

da particolari agenti esterni a priori magari non considerati (come ad esempio un rumore esterno, un cambio di impugnatura, ecc.).

3.4.1 Questionario

Ai soggetti che si sono offerti per sostenere il test è stato chiesto loro di compilare un questionario perchè le loro risposte potessero essere un'utile integrazione da prendere in considerazione durante l'analisi dei dati. Le domande presentate nel seguente questionario riguardano:

- Dati anagrafici:
 - Nome
 - Cognome
 - Data di nascita
 - Luogo di nascita
 - Luogo di residenza
- Dati legati a possibili problemi fisici:
 - Problemi di vista specificando anche l'entità
 - Porta lenti o occhiali
 - Problemi di udito
 - Problemi all'arto superiore usato per il test
- Dati legati al livello di manualità:
 - Braccio dominante
 - Sport praticati e rispettive ore di attività sportiva settimanale
 - Strumenti suonati e rispettive ore di pratica giornaliera
 - Ore giornaliere passate per giocare alla consolle e tipo di joystick usato
 - Ore giornaliere passate davanti a TV o altro monitor per attività di gioco e tipo di joystick usato

- Ore giornaliere passate davanti ad un monitor per attività di sola navigazione personale
- Ore giornaliere passate davanti ad un monitor per attività lavorative o di studio

Dal questionario in questione siamo stati in grado di reperire alcune informazioni da verificare successivamente se possono essere utili ai fini dell'analisi dei dati.

3.4.2 Gruppi ottenuti mediante randomizzazione

I soggetti reperiti per il test sono stati suddivisi mediante un programma di randomizzazione con il quale si è cercato nel miglior modo possibile di distribuire tra i 4 gruppi i maschi e le femmine o i soggetti mancini e destrorsi. I 4 gruppi generati presentano le seguenti caratteristiche:

1. Gruppo1: NF (NoFeedback)

Composto da 11 soggetti, 2 femmine e 9 maschi, tutti destrorsi. L'età media di tale gruppo è 23,69 anni e tutti i componenti hanno un'età compresa tra i 22 e i 26 anni. Nessuno dei soggetti appartenenti a tale gruppo presenta rilevanti caratteristiche che possano influenzare la performance del test.

2. Gruppo2: APER (AudioPositionErrorRelated)

Composto da 12 soggetti, 3 femmine e 9 maschi, tutti destrorsi tranne uno che è mancino. L'età media di tale gruppo è 24,95 anni e tutti i componenti hanno un'età compresa tra i 22 e i 37 anni. Nessuno dei soggetti appartenenti a tale gruppo presenta rilevanti caratteristiche che possano influenzare la performance del test.

3. Gruppo3: ATRV (AudioTaskRelated_scalaVideo)

Composto da 12 soggetti, 3 femmine e 9 maschi, tutti destrorsi tranne uno che è mancino. L'età media di tale gruppo è 23,85 anni e tutti i componenti hanno un'età compresa tra i 22 e i 26 anni. Nessuno dei soggetti appartenenti a tale gruppo presenta rilevanti caratteristiche che possano influenzare la performance del test.

4. Gruppo4: ATRJ (AudioTaskRelated_scalaJoystick)

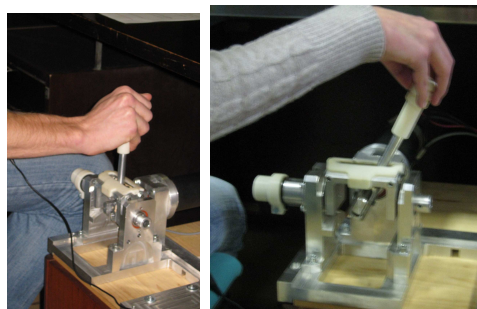
Composto da 12 soggetti, 2 femmine e 10 maschi, tutti destrorsi. L'età media di tale gruppo è 25,33 anni e tutti i componenti hanno un'età compresa tra i 22 e i 31 anni. Nessuno dei soggetti appartenenti a tale gruppo presenta rilevanti caratteristiche che possano influenzare la performance del test.

3.4.3 Impugnature

Nonostante fosse stato chiesto al soggetto di prendere confidenza con il joystick durante la "Fase 0" di modo che ognuno potesse ipotizzare quale fosse l'impugnatura più conveniente da mantenere durante l'intera esecuzione del test, è stato osservato che più di un soggetto, consapevolmente o meno, ha cambiato l'impugnatura scelta. Nell'analizzare i dati, in corrispondenza di questa variazione di impugnatura, è stato possibile notare in modo evidente tale cambiamento. È stata presa nota delle diverse impugnature scelte dai soggetti e nel complesso sono riconducibili a 5 diverse tipologie per praticità denominate:

1. **PC**: Il joystick viene manovrato impugnandolo(Vedi Fig.3.7(a))
2. **PCD**: Il joystick viene manovrato con tutte le dita(Vedi Fig.3.7(b))
3. **2D**: Il joystick viene manovrato con solo due dita
4. **PALMO**: Il joystick viene manovrato con il palmo
5. **2D+Braccio**: Il joystick viene manovrato con solo due dita, ma il braccio dominante era guidato dall'altro braccio

Nelle figure subito a seguito sono raffigurate due tipologie diverse di impugnatura riscontrate in due diversi soggetti sottoposti al test:



(a) Impugnatura PC (b) Impugnatura PCD

Figura 3.7: Due diverse tipologie di impugnatura

3.5 Problemi riscontrati

Ancor prima di sottoporre i soggetti al test sviluppato nel seguente progetto, si è provato a pensare quali potevano essere i problemi che nel decorso dei test sarebbero potuti essere compromettenti per la corretta valutazione degli esercizi. Nel riflettere su questo, abbiamo notato che l'utilizzo del joystick come strumento di lavoro installato su una cassettera di legno sarebbe potuto essere scomodo da utilizzare da parte di tutti coloro che erano mancini. Infatti prima di eseguire il test: ai destrorsi la cassettera con il joystick è stata posizionata alla loro destra, mentre per i mancini è stata posizionata di fronte a loro.

Nel caricare il programma elaborato i problemi principali riscontrati erano legati alla visualizzazione dei target a video i quali spesso risultavano tagliati ma mai di un quantitativo superiore al 50% della loro dimensione totale. Un ulteriore problema riscontrato per tempo, ovvero prima di sottoporre i soggetti al test, è inerente al fatto che spesso Windows era vittima di crash. Le ipotesi più plausibili per valutare la causa di tale effetto sono legate al fatto che il software Simulink nel generare un modello Real-Time assume la massima priorità per quanto riguarda le operazioni da svolgere. Gli altri programmi in esecuzione assumono una priorità praticamente nulla. Ecco perchè Matlab nel caricare i valori pervenuti dal modello Simulink non riesce a gestire la cosa con i tempi pilotati da Simulink stesso e si presume che questo porti ad un crash di Windows.

Dal punto di vista dei soggetti sottoposti al test una delle problematiche maggiori che hanno in molti evidenziato era legata alla poca ergonomia della leva del joystick che per molti è stata una delle cause che ha provocato un aumento dell'errore relativo alla performance.

Test Variable Amplitude - GRUPPO 1: NF-No Feedback

Nome	Sesso	Braccio	Occhiali (O) o Lenti (L)	Data Test	Ora Test	Data di nascita	Età
Turella Elisabetta	F	DX	L	18/01/2011	14.00 - 14.20	19/08/1988	22,51
Bianchi Giulia	F	DX	L	17/01/2011	10.20 - 10.40	26/08/1987	23,50
Borgato Alessandro	M	DX	L	18/01/2011	11.00 - 11.20	22/08/1988	22,50
Campion Andrea	M	DX	L	17/01/2011	16.40 - 17.00	07/01/1986	25,13
Salata Adriano	M	DX	X	19/01/2011	11.00 - 11.20	11/04/1988	22,87
D'Onofrio Giuseppe	M	DX	X	18/01/2011	13.20 - 13.40	20/01/1987	24,09
Callegaro Daniel	M	DX	X	17/01/2011	14.00 - 14.20	16/08/1988	22,52
Frezzato Nicolò	M	DX	X	19/01/2011	15.30 - 15.50	11/09/1987	23,45
Finetto Christian	M	DX	X	19/01/2011	17.35 - 17.55	08/11/1986	24,29
Dal Corso Nicola	M	DX	X	20/01/2011	12.40 - 13.00	18/10/1984	26,35
Pasetto Damiano	M	DX	X	20/01/2011	09.40 - 10.00	30/08/1987	23,48

Numero elementi appartenenti al gruppo

— 11

Età media del gruppo

— 23,70

Tabella 3.1: Tabella caratteristiche soggetti reclutati

Test Variable Amplitude - GRUPPO 2: APER-Audio Position Error Related

Nome	Sesso	Braccio	Occhiali (O) o Lenti (L)	Data Test	Ora Test	Data di nascita	Età
Bovo Antonio Maria	M	DX	X	18/01/2011	09.40 - 10.00	09/12/1988	22,21
Durigon Alberto	M	SX	O	17/01/2011	10.00 - 10.20	15/12/1987	23,19
Fetovsky Zahir	M	DX	X	18/01/2011	09.30 - 09.40	26/09/1988	22,41
Bonutto Dimitri	M	DX	L	17/01/2011	12.00 - 12.20	28/05/1988	22,74
Marcato Alice	F	DX	X	20/01/2011	19.20 - 19.40	07/08/1988	22,55
Callegari Gianluca	M	DX	X	18/01/2011	12.20 - 12.40	14/02/1987	24,02
Nalin Chiara	M	DX	X	17/01/2011	13.40 - 14.00	14/10/1987	23,36
Caruso Marco	M	DX	X	19/01/2011	16.40 - 17.00	15/01/1985	26,11
Tonazzo Marco	M	DX	O	19/01/2011	09.00 - 09.30	13/02/1987	24,03
De Luca Jessika	F	DX	X	20/01/2011	12.20 - 12.40	21/12/1986	24,18
Rosati Giulio	M	DX	L	20/01/2011	18.20 - 18.40	05/07/1973	37,65
Oscari Fabio	M	DX	X	20/01/2011	16.00 - 16.20	23/01/1984	27,09

Numero elementi appartenenti al gruppo

— 12

Età media del gruppo

— 24,96

Tabella 3.2: Tabella caratteristiche soggetti reclutati

Test Variable Amplitude - GRUPPO 3: ATRV-Audio Task Related Video

Nome	Sesso	Braccio	Occhiali (O) o Lenti (L)	Data Test	Ora Test	Data di nascita	Età
Campa Andrea	M	DX	X	18/01/2011	11.20 - 11.40	25/01/1988	23,08
Caramma Lea	F	DX	X	20/01/2011	10.00 - 10.20	26/11/1988	22,24
Roncagalli Cassandra	F	DX	X	18/01/2011	14.40 - 15.00	01/05/1988	22,81
Milani Bruno	M	SX	X	18/01/2011	13.00 - 13.20	23/08/1984	26,50
Momolo Alessandro	M	DX	O	19/01/2011	11.20 - 11.40	17/05/1988	22,77
Melan Giacomo	M	DX	X	18/01/2011	17.20 - 17.40	13/04/1987	23,87
Perini Claudia	F	DX	X	18/01/2011	11.40 - 12.00	09/12/1988	22,21
Boschetto Davide	M	DX	X	18/01/2011	17.40 - 18.00	22/01/1987	24,09
Minto Simone	M	DX	X	19/01/2011	17.00 - 17.20	17/10/1984	26,35
Stecca Filippo	M	DX	O	19/01/2011	09.30 - 09.50	29/03/1986	24,91
Desiderà Alberto	M	DX	O	20/01/2011	09.00 - 09.20	12/06/1987	23,70
Battistella Massimo	M	DX	X	20/01/2011	15.00 - 15.20	04/06/1987	23,72

Numero elementi appartenenti al gruppo

— 12

Età media del gruppo

— 23,85

Tabella 3.3: Tabella caratteristiche soggetti reclutati

Test Variable Amplitude - GRUPPO 4: ATRJ-Audio Task Related Joystick

Nome	Sesso	Braccio	Occhiali (O) o Lenti (L)	Data Test	Ora Test	Data di nascita	Età
Sattin Mattia	M	DX	X	17/01/2011	16.00 - 16.20	03/08/1985	25,56
Zanata Chiara	F	DX	X	17/01/2011	16.40 - 17.00	06/10/1988	22,38
Zampicinini Andrea	M	DX	X	19/01/2011	12.00 - 12.20	19/08/1980	30,52
Rampado Andrea	M	DX	X	19/01/2011	12.40 - 13.00	22/06/1984	26,67
Sartorato Paolo	M	DX	O	18/01/2011	09.00 - 09.20	23/11/1987	23,25
Smania Giovanni	M	DX	X	18/01/2011	17.00 - 17.20	12/10/1987	23,37
Roncali Loris	M	DX	X	20/01/2011	09.20 - 09.40	23/08/1988	22,50
Santuliana Marco	M	DX	X	17/01/2011	14.20 - 14.40	21/05/1986	24,76
Vlad Andrei	M	DX	X	17/01/2011	14.40 - 15.00	30/03/1986	24,90
Beghetto Giulia	F	DX	O	19/01/2011	09.50 - 10.10	04/11/1986	24,30
Cazzaro Alberto	M	DX	O	19/01/2011	17.15 - 17.35	18/10/1986	24,35
Secoli Riccardo	M	DX	X	20/01/2011	15.40 - 16.00	28/08/1979	31,50

Numero elementi appartenenti al gruppo

— 12

Età media del gruppo

— 25,34

Tabella 3.4: Tabella caratteristiche soggetti reclutati

Capitolo 4

Analisi Statistica

In questo capitolo vengono presentate l'elaborazione dei dati e l'analisi statistica effettuata al fine di valutare l'efficacia dei feedback progettati. Per com'è stato strutturato il test *Variable Amplitude* l'analisi in questione si basa su un confronto incrociato fra le quattro diverse tipologie di gruppi creati. Dal punto di vista esecutivo, come spiegato in precedenza, la tipologie di esercizio eseguito da ciascun soggetto era la medesima. L'appartenenza a gruppi differenti è legata alla tipologia di audio con il quale ciascun elemento ha sostenuto le prove di 'target tracking'. Per consentire un'analisi statistica dettagliata si andrà a calcolare, per ogni soggetto, una serie di parametri i cui valori daranno un'indicazione globale sulla qualità con cui è stato svolto il task stesso. Sull'entità e sul segno della differenza data dai valori di questi parametri si basa l'analisi statistica effettuata.

4.1 Parametri calcolati per l'elaborazione

Di seguito vengono definiti e brevemente descritti i tipi di parametri considerati nell'analisi statistica dei test. Molti di questi errori, in particolare tutti quelli riconducibili a distanze o differenze di posizione, sono stati normalizzati sulla dimensione del raggio del target di modo che il risultato numerico sia di più facile ed immediata interpretazione. Le formule riportate riguardano il calcolo dell'errore in un generico tratto della traiettoria.

4.1.1 Distanza media dal target (dism)

Per distanza media si intende la media delle distanze, in ogni istante di campionamento, fra l'indicatore del target e quello del soggetto. La distanza va intesa come distanza sul piano (distanza euclidea) e pertanto tiene conto sia della differenza di coordinate in x che quella in y. In formule:

$$\epsilon = \frac{1}{N} \sum_{k=1}^N \frac{\sqrt{(x_s(t) - x_t(t))^2 + (y_s(t) - y_t(t))^2}}{R} \quad (4.1)$$

dove N indica il numero di campioni considerati, e R un fattore di normalizzazione pari al raggio del target.

L'interpretazione di tale errore è immediata, inoltre si ricordi il risultato che fornisce questo parametro non potrà mai essere negativo e sarà indipendente dal segno delle differenze di posizione lungo gli assi x e y, cioè tratterà allo stesso modo equivalenti condizioni di ritardo o anticipo del soggetto rispetto al target.

4.1.2 Integrale del modulo della velocità relativa (dtt)

Questo tipo di errore valuta per ogni ciclo l'integrale dato dai campioni del modulo della velocità relativa fra soggetto e target. L'idea è quella di valutare la quantità:

$$\int_{t_1}^{t_2} |\vec{v}_s - \vec{v}_t| dt \quad (4.2)$$

La valutazione dell'integrale avviene in maniera approssimata sommando i contributi relativi ad ogni campione pesato con l'intervallo di campionamento (0,01s). Il calcolo di questo parametro di errore consiste nell'implementazione del metodo dei rettangoli per il calcolo numerico dell'integrale:

$$\epsilon = \sum_{k=1}^N \frac{\sqrt{(v_{x,s}(t) - v_{x,t}(t))^2 + (v_{y,s}(t) - v_{y,t}(t))^2} dt}{|x_t(N) - x_t(1)|} \quad (4.3)$$

dove N rappresenta il numero di campioni del ciclo, mentre il fattore di normalizzazione presente al denominatore rappresenta la lunghezza del tratto percorso nel ciclo considerato. Tale parametro misura lo spazio percorso dal soggetto avente modulo del vettore velocità pari a $(v_s - v_t)$ e pertanto da un'indicazione sullo spazio percorso inutilmente in ogni ciclo, indipendentemente dall'accuratezza

con cui l'esercizio è stato eseguito. Il risultato fornito dall'integrale viene scalato sulla lunghezza del tratto percorso dal target.

4.1.3 Ritardo medio (emx)

L'errore di posizione lungo l'asse x di ogni singolo campione è definito come:

$$(x_s - x_t) \text{sign}(v_{x_t}) \quad (4.4)$$

dove la differenza delle posizioni lungo l'asse x fra soggetto e target ($x_s - x_t$) è pesata con la direzione del movimento del target, ovvero con il segno della velocità di quest'ultimo. L'errore di posizione medio è dato dunque dalla formula:

$$\epsilon = \frac{1}{N} \sum_{k=1}^N \frac{(x_s(t) - x_t(t)) \text{sign}(v_{x_t}(t))}{R} \quad (4.5)$$

dove N indica il numero di campioni considerati e r il fattore di normalizzazione pari alla dimensione del raggio dell'indicatore del target.

A questo tipo di errore ci si può riferire anche con il termine ritardo medio poichè si è visto che, mediamente, tale errore in tutti i task e per tutti i soggetti risulta essere negativo, mettendo in luce il fatto che nell'inseguimento da parte di un soggetto sono più comuni le situazioni di ritardo rispetto a quelle di anticipo, come del resto è lecito attendersi.

Tale errore, tuttavia, si è rivelato inconsistente nel momento in cui avviene il cambio di direzione da parte del target, ed in conseguenza del segno della sua velocità: in generale il soggetto tende ad essere in ritardo rispetto al target, pertanto, quando quest'ultimo ha già completato il proprio ciclo, il soggetto deve ancora portarlo a termine; poichè l'errore di posizione calcolato in 4.5 tiene conto solo del segno della velocità del target, nel momento in cui avviene il cambio del ciclo, il soggetto, che era in ritardo rispetto al target, viene ora considerato in anticipo. Tale situazione si protrae finchè i segni delle velocità dei 2 indicatori rimangono discordi. Per ovviare a tale inconveniente è stato introdotto un fattore correttivo, applicato nella prima metà di ogni ciclo:

$$(-1)^c \quad (4.6)$$

dove c risulta pari a:

$$(v_{x_s}(t) - v_{x_t}(t)) \text{sign}(x_s(t) - x_t(t)) < 0 \wedge v_{x_s}(t) \neq v_{x_t}(t) \wedge |v_{x_t}(t) - v_{x_t}(t-1)| > 0 \quad (4.7)$$

dove:

- $(v_{x_s}(t) - v_{x_t}(t)) \text{sign}(x_s(t) - x_t(t)) < 0$: stabilisce se il soggetto è in ritardo rispetto al target nell'istante in cui quest'ultimo cambia ciclo
- $v_{x_s}(t) \neq v_{x_t}(t)$: verifica, in base ai segni discordi delle velocità, che i 2 indicatori abbiano direzioni opposte
- $|v_{x_t}(t) - v_{x_t}(t-1)| > 0$: stabilisce se il modulo della velocità del target è crescente; in tal modo viene considerata solo la prima metà del ciclo.

L'operazione di elevamento a potenza restituisce -1 quando tutte le condizioni sono verificate simultaneamente, restituendo un lag error quando si tratta effettivamente di un ritardo.

4.1.4 Errore medio di posizione in y (emy)

L'errore di posizione lungo l'asse y di ogni singolo campione è definito come:

$$\epsilon = \frac{1}{N} \sum_{k=1}^N \frac{(x_s(t) - x_t(t))}{R} \quad (4.8)$$

dove N indica il numero di campioni considerati e R il fattore di normalizzazione pari alla dimensione del raggio dell'indicatore del target.

Di fatto, tale errore corrisponde alla posizione del soggetto lungo l'asse y, in quanto la posizione del target è sempre nulla in tale direzione.

4.1.5 Errore quadratico medio in x (eqmx)

L'errore quadratico medio è la media della distanza, considerata solo lungo l'asse x, in ogni istante di campionamento, fra l'indicatore del target e quello del soggetto. In formule risulta:

$$\epsilon = \frac{1}{R} \sqrt{\frac{\sum_{k=1}^N (x_s(t) - x_t(t))^2}{N}} \quad (4.9)$$

dove N indica il numero di campioni considerati e R il fattore di normalizzazione pari alla dimensione del raggio dell'indicatore del target.

4.1.6 Errore quadratico medio in y (eqmy)

L'errore quadratico medio è la media della distanza, considerata solo lungo l'asse x , in ogni istante di campionamento, fra l'indicatore del target e quello del soggetto. In formule risulta:

$$\epsilon = \frac{1}{R} \sqrt{\frac{\sum_{k=1}^N (y_s(t) - y_t(t))^2}{N}} \quad (4.10)$$

al solito, N indica il numero di campioni considerati e R il fattore di normalizzazione pari alla dimensione del raggio dell'indicatore del target.

4.1.7 Lead error (lead)

Si definisce lead error la media fra i soli contributi positivi dell'errore di posizione medio, in formule risulta:

$$\epsilon = \frac{1}{N} \sum_{k=1}^N \frac{(err_x(t))(sign(err_x(t)) > 0)}{R} \quad (4.11)$$

dove err_x indica la quantità 4.4 con il termine correttivo. Tale parametro d'errore indica qual è mediamente l'errore di posizione lungo l'asse x nei soli istanti di tempo in cui il soggetto anticipa il target nel movimento.

4.1.8 Lag error (lag)

Il lag error è il complementare del lead error rispetto all'errore di posizione. Infatti, si definisce lag error la media fra i soli contributi negativi dell'errore di posizione medio; in formule risulta:

$$\epsilon = \frac{1}{N} \sum_{k=1}^N \frac{(err_x(t))(sign(err_x(t)) < 0)}{R} \quad (4.12)$$

Tale parametro indica qual è mediamente l'errore di posizione nei soli istanti di tempo in cui il soggetto posticipa il target nel movimento.

4.2 Analisi statistica

Per l'analisi statistica sui dati relativi ai test si è scelto di sostituire due dei soggetti che hanno partecipato all'esercizio a causa dell'elevata distrazione con i quali si sono sottoposti all'esercizio. Fra le 47 persone testate si è scelto di scartare nell'elaborazione 5 soggetti rispettivamente 3 dal gruppo ATRV e 2 dal gruppo ATRJ a causa della. Questa scelta è motivata soprattutto dalla richiesta di poter avere a disposizione la maggiore quantità di confronti possibili fra i vari task.

4.2.1 Test di Mann-Whitney

Il Wilcoxon test è un test non parametrico che confronta la mediana di una colonna di numeri con una mediana ipotetica. Il responso di tale confronto viene dato mediante p-value. Il p-value risponde a questa domanda: se i dati sono stati raccolti da una popolazione con una mediana pari all'ipotetico valore inserito, qual è la probabilità di selezionare casualmente N campioni e trovare una mediana il più lontano possibile dal valore osservato? Per l'analisi statistica vi sono delle deduzioni legate al p-value. Se il p-value è piccolo, è possibile rifiutare l'idea che la differenza sia dovuta al caso e concludere, invece, che la popolazione ha una media distinta dal valore ipotetico immesso. Se il p-value è elevato, i dati non hanno nessuna ragione di concludere che la mediana della popolazione differisca dalla mediana ipotetica. Questo non è lo stesso che affermare che le mediane sono le stesse, non si ha semplicemente nessuna prova che queste sia diverse. Se si dispone di piccoli campioni, il Wilcoxon test non è molto efficace: infatti, se si dispone di 5 valori, o anche meno, il Wilcoxon test darà sempre un p-value maggiore di 0,05, indipendente da quanto la mediana del campione disti da quella ipotetica. Il Wilcoxon test non assume che i dati siano campionati con una distribuzione gaussiana ma che siano distribuiti simmetricamente intorno alla mediana. Se la distribuzione è asimmetrica, il p-value non dirà molto sul fatto che la mediana sia o meno diversa dal valore ipotetico. Come tutti i test statistici, il Wilcoxon considera gli errori indipendenti, con il cui termine si fa riferimento alla differenza tra ogni valore e la mediana. Il risultato del Wilcoxon test hanno senso solo se la

dispersione è casuale, cioè che un fattore che causa un valore troppo alto o troppo basso interessi solo per quel dato valore.

4.2.2 Unpaired t-test

L'Unpaired t-test confronta le medie di due gruppi non assortiti, supponendo che i valori assumano una distribuzione gaussiana. Il risultato più utile di tale test corrisponde all'intervallo di confidenza nel quale è possibile trovare il valore di differenza fra le medie dei due gruppi. Se le assunzioni sulle analisi fatte sono corrette, è possibile avere una sicurezza del 95% che l'intervallo di confidenza considerato conterrà il vero valore di differenza tra le medie dei gruppi. Il p-value considerato per l'analisi statistica in questione viene utilizzato per osservare se la differenza tra la media dei due gruppi è in base ad una analisi reale o è dovuta al caso. Il p-value risponde ad una domanda: se le due popolazioni hanno veramente la stessa media, qual'è la probabilità che campioni presi a caso abbiano la stessa media come osservato in questo esperimento?

In base al p-value le considerazioni da fare possono essere svariate: un piccolo valore di p significa che l'associazione che è stata osservata sarebbe raro accadesse nel caso di un campione casuale. Un valore inferiore allo 0,05 per cento indica che l'intervallo di confidenza al 95% non conterrà zero quando si confrontano le due medie. Un valore grande di p ci permette di dire che non abbiamo nessuna ragione per concludere che le medie differiscono. Anche se le medie reali erano uguali, non ci sarebbe stato nulla di sorprendente nel trovare a caso valori di medie distanti.

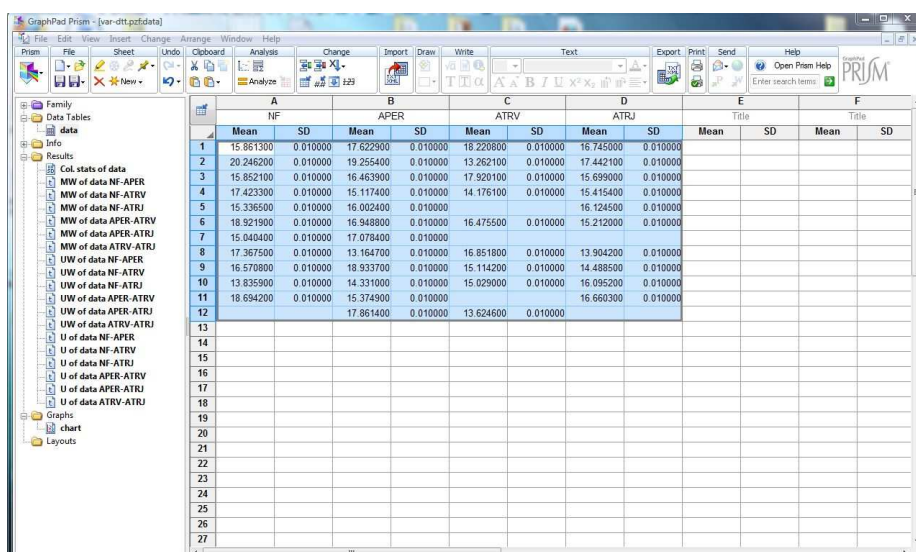
4.2.3 Unpaired t-test con Welch's correction

Nei t-test spaiati si presuppone che le due popolazioni siano caratterizzate dalla medesima varianza. Dal momento che la varianza è uguale alla deviazione standard al quadrato, questo significa che le popolazioni hanno la stessa deviazione standard. Una modifica del t-test (sviluppato da Welch) può essere utilizzato quando si è disposti a fare questo assunto. Questa scelta è disponibile solo per gli unpaired t-test. Con il t-test di Welch, i gradi di libertà sono calcolati da

una equazione complicata e il numero non è ovviamente correlato alla dimensione del campione. T test di Welch è usato raramente.

4.3 Software per l'analisi statistica: GraphPad Prism

Per le conclusioni statistiche sui dati memorizzati si è fatto ricorso al software specifico GraphPad Prism. GraphPad Prism è un software statistico che combina grafica scientifica, statistiche e curve di interpolazione in un programma molto semplice da utilizzare. Tramite questo programma si possono creare diverse fun-



	A NF		B APER		C ATRV		D ATRJ		E Title		F Title	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
1	15.861300	0.010000	17.622900	0.010000	18.220800	0.010000	16.744500	0.010000				
2	20.246200	0.010000	19.255400	0.010000	13.262100	0.010000	17.442100	0.010000				
3	15.852100	0.010000	16.463900	0.010000	17.920100	0.010000	15.699000	0.010000				
4	17.423300	0.010000	15.117400	0.010000	14.176100	0.010000	15.415400	0.010000				
5	15.336500	0.010000	16.002400	0.010000			16.124500	0.010000				
6	18.921900	0.010000	16.948800	0.010000	16.475500	0.010000	15.212000	0.010000				
7	15.040400	0.010000	17.078400	0.010000								
8	17.367500	0.010000	13.164700	0.010000	16.851800	0.010000	13.904200	0.010000				
9	16.570800	0.010000	18.933700	0.010000	15.114200	0.010000	14.488500	0.010000				
10	13.835900	0.010000	14.331000	0.010000	15.029000	0.010000	16.095200	0.010000				
11	18.694200	0.010000	15.374900	0.010000			16.660300	0.010000				
12			17.861400	0.010000	13.624600	0.010000						
13												
14												
15												
16												
17												
18												
19												
20												
21												
22												
23												
24												
25												
26												
27												

Figura 4.1: Prism: software utilizzato per l'analisi statistica

zioni statistiche in grado di aggiornarsi istantaneamente nel momento in cui si va a cambiare i dati dalla tabella di origine. Lo stesso concetto vale anche per la creazione dei grafici. Per l'analisi in questione si è scelto di prendere in considerazione il valor medio e la deviazione standard per ciascuno degli 8 parametri calcolati.

4.3.1 Procedimento

Per l'analisi in questione sono stati sfruttati dei file Matlab messi a disposizione i quali, mediante qualche modifica, potevano essere adattati al caso nostro.

- *SaveDat.m*: file mediante il quale è possibile a partire dai file *.mat* (ovvero i file ottenuti dall'esecuzione degli esercizi svolti dai soggetti) ricavare file *.dat* che sono file nei quali vengono rispettivamente calcolati tutti i parametri definiti in precedenza (*distm*, *dtm*, *emx*, *emy*, *eqmx*, *eqmy*, *lead* e *lag*). Le misure vengono effettuate una prima volta considerando la durata totale dell'esercizio, mentre poi vengono mediate ciclo per ciclo. I valori calcolati vengono rispettivamente salvati in strutture chiamate *data* e *datac*
- *SaveDatTaroc.m*: file mediante il quale è possibile eliminare da alcuni soggetti solo i cicli definiti "fuori norma", ovvero quei cicli isolati rispetto la prestazione dell'intero esercizio, dovuti probabilmente da istanti di distrazione del soggetto. Nel nostro caso specifico i cicli anomali non vengono eliminati bensì si è scelto di copiare i dati del ciclo precedente al ciclo anomalo, sul ciclo anomalo, in modo da avere una caratteristica più omogenea prima del picco fuori norma. Successivamente le strutture *data* e *datac* vengono aggiornate
- *xlsDatAmplitude.m*: tale file è in grado di aprire tutti i file *.dat* precedentemente creati, e in base alle impostazioni con il quale è stato definito calcola le performance del soggetto (media e deviazione standard) relative a ciascun parametro considerato. Una volta eseguiti i calcoli salva i dati ottenuti su file *.xls* utili per Prism. Viene inoltre generato il file *settings.mat* per tenere traccia dei parametri impostati al lancio del file
- *xlsDatAmplitude.m*: *pScript.pzc*, implementato in Prism, importa i valori dei p-value generati da ogni confronto nei tre file *MW.csv* e *U.csv* e *UW.csv* rispettivamente del Mann-Whitney, dell'Unpaired t-test e dell'Unpaired t-test con Welch's corrections. Inoltre genera il file *p-values-amplitude.xlsx* che contiene il riferimento ai tre file appena nominati

- i quattro file (MW, U, UW e p-values-amplitude) devono essere aperti contemporaneamente: il file p-values-color.xlsx contiene tre fogli che vengono aggiornati automaticamente prendendo i contenuti di MW.csv, U.csv e UW.csv, pertanto è necessario salvarlo prima di ogni chiusura
- *saveResAmplitude.m* salva tutti i grafici di Prism e le tabelle Excel di ogni parametro, il file settings.mat e il file.xlsx del p-value in una sottocartella della cartella 'confronti'

4.4 Risultati

In questo paragrafo vengono presentati i primi risultati reperiti dall'elaborazione, che faranno riferimento solo all'ultima delle tre fasi, ovvero alla fase si Var_Test, rinviando lo studio approfondito sulle prime due parti a colei che porterà avanti questo progetto. Per quanto possano essere dei risultati parziali perchè riferiti ad una sola porzione dell'esercizio proposto, si cerca di darne una concreta interpretazione.

4.4.1 Distanza dal target: dtt

Se si osservano i grafici risultanti dall'elaborazione sul parametro *dtt* (distanza dal target), si può notare che indipendentemente dal gruppo di appartenenza, ovvero dalla tipologia di feedback audio applicato, i valori risultano praticamente equiparabili. La prima conclusione che si può trarre da questo, ricollegandosi anche a studi precedentemente eseguiti su esercizi di target tracking simili (Figura 4.2), è che l'esercizio proposto risulta essere caratterizzato da un livello di difficoltà elevato. Probabilmente l'inseguimento del target da parte del soggetto era già sufficientemente complicato con una traiettoria variabile da parte del target di riferimento, e l'aggiunta della variazione di scala ha fatto sì che l'errore del soggetto aumentasse (come evidenziato dalla figura).

Riferendoci nello specifica ai soli grafici riferiti al Var_Test di Variable Amplitude possiamo notare come non vi siano rilevanti differenze tra i grafici di un gruppo rispetto ai grafici di un altro gruppo, ciò significa che dal punto di vista

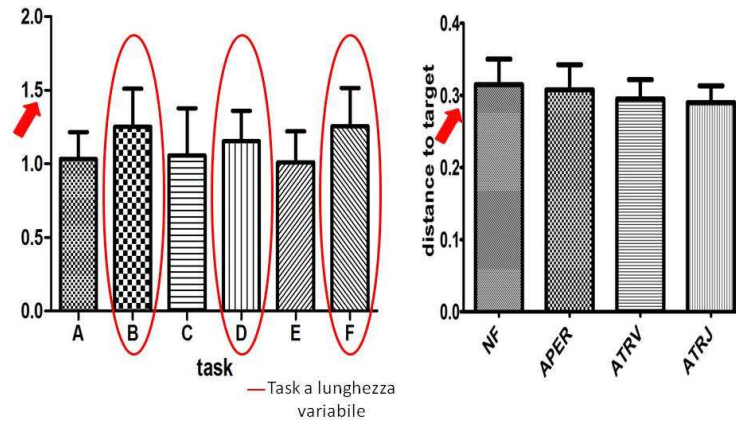


Figura 4.2: Confronto fra dtt della fase Var_Test di Variable Amplitude (a destra) con esercizi di target tracking svolti precedentemente (a sinistra) e caratterizzati da sola lunghezza variabile senza variazione di scalatura

del feedback audio applicato, nessuna delle quattro tipologie ha aiutato il soggetto a migliorare la sua performance.

L'analisi statistica eseguita conferma ciò che è appena stato affermato: nelle tabelle dei pValue riportate in seguito è possibile notare come mettendo a confronto i diversi gruppi nessuno di essi evidenzia valori di pValue rilevanti, ciò sta a significare che dal punto di vista statistico i diversi confronti non hanno mostrato prevalenza dell'effetto migliorativo di un feedback audio rispetto ad un altro.

4.4.2 Distanza media dal target: *dism*

Dai grafici riferiti a *dism*, considerata come la media delle distanze fra il target di riferimento e quello rappresentativo del soggetto in ogni istante di campionamento, risulta evidente come i gruppi 3 e 4, rispettivamente chiamati ATRV e ATRJ, sono caratterizzati da un'accuratezza più elevata durante la fase di target tracking rispetto al gruppo 2 chiamato APER. A differenza se paragoniamo il gruppo 2 (APER) al gruppo 1 (NF) non si evidenziano particolari vantaggi legati alla tipologia di audio in questione, anzi sembra che l'audio legato all'er-

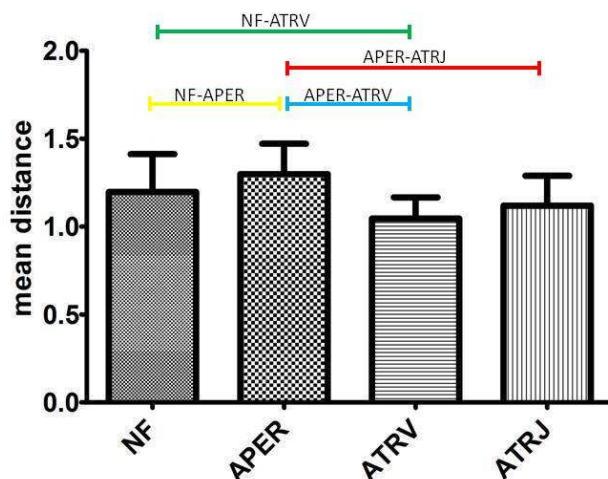


Figura 4.3: Grafico riferito alla distanza media dal target da parte dei soggetti nella fase Var_Test di Variable Amplitude

rore di posizione possa causare un aumento del valore di distm. A sostegno di tale affermazione basta riferirsi alle tabelle sui pValue rappresentate in seguito. È evidenziato come i confronti fra il gruppo APER con i due rispettivi gruppi ATRV e ATRJ risultino anche statisticamente rilevanti. Nonostante non superi la soglia statistica imposta sul valore di pValue (< 0.05) si può osservare che il confronto fra i gruppi ATRV ed NF porta a valori che molto si avvicinano alla soglia di rilevanza statistica. In conclusione è possibile affermare che in base ai grafici ottenuti il gruppo sottoposto a feedback audio relativo all'errore di velocità del target con applicato alla traiettoria video del target (ATRV) e quello sottoposto a feedback audio relativo all'errore di velocità del target applicato alla traiettoria tracciata con il joystick (ATRJ) siano di maggior aiuto per il soggetto nell'eseguire gli esercizi di target tracking.

4.4.3 Errore medio in x : emx

Prendendo in considerazione i grafici di emx considerato come l'errore medio commesso rispetto all'asse x , definito in altri termini "ritardo medio in x ", risulta evidente come il gruppo 2 denominato APER ha dei risultati nettamente peggiori rispetto agli altri tre gruppi. Ad avvalorare la tesi basti osservare i relativi pValue indicati nei confronti fra il gruppo APER e rispettivamente, i gruppi NF, ATRJ e ATRV. Quindi osservando i grafici relativi all'errore medio in x balza all'occhio che il gruppo APER è caratterizzato da un valore medio d'errore più elevato rispetto agli altri tre gruppi e tale ipotesi è quindi sostenuta dall'analisi statistica effettuata. Considerando quindi i grafici inerenti ad emx come degli indici

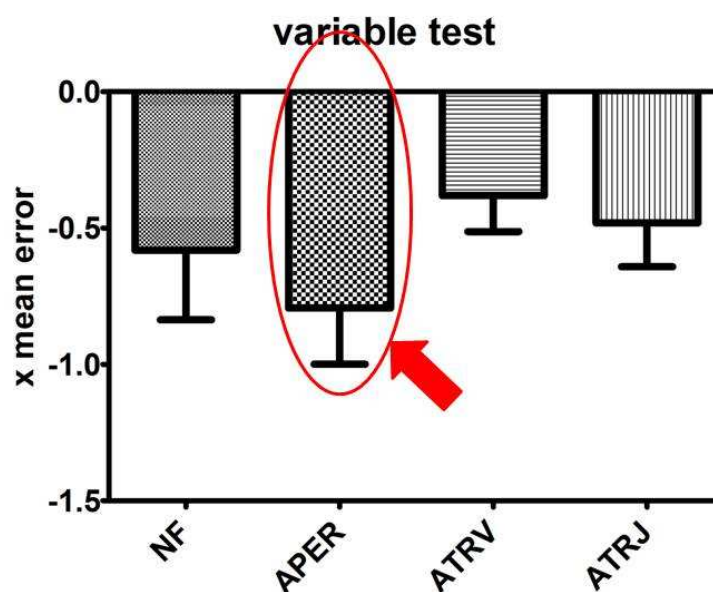


Figura 4.4: Grafico riferito al ritardo medio in x nella fase Var_Test di Variable Amplitude

riguardanti le performance del soggetto, notiamo che l'audio relativo all'errore di posizione del soggetto rispetto al target di riferimento risulta essere quello che causa maggior ritardo rispetto agli altri tre. Non è comunque da dimenticare un

concetto già espresso in precedenza: la tipologia di test caratterizzata da una lunghezza variabile del target a video e una scalatura legata al movimento del joystick, è comunque da considerarsi un esercizio molto difficoltoso da svolgere. Per rafforzare tale concetto si basti osservare i grafici relativi all'errore quadratico medio. Di rilevanza statistica risultano i confronti tra il gruppo APER e i due gruppi ATRV e ATRJ, mentre il confronto tra il gruppo APER e il gruppo NF è molto vicino alla soglia statistica considerata.

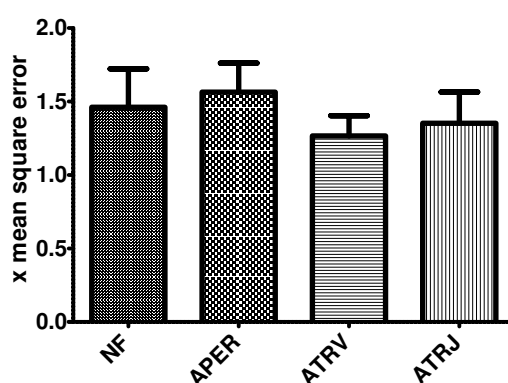


Figura 4.5: Grafico riferito all'errore quadratico medio in x nella fase Var_Test di Variable Amplitude

Confronto di emx con test precedenti

Confrontando il grafico di emx riferito ai test precedentemente eseguiti con il grafico di emx ricavato per il test trattato (Variable amplitude), si possono notare alcuni aspetti molto interessanti. Essendo anche i test precedenti degli esercizi di target tracking nei quali il soggetto doveva sempre seguire un target che descriveva una traiettoria variabile (il soggetto non era però vittima di cambio scale), confrontando i risultati si può notare che:

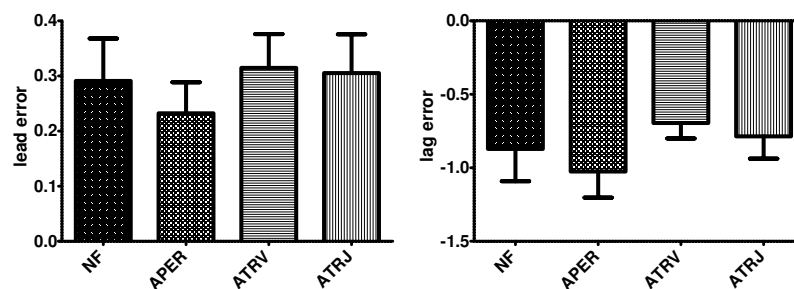
1. nei test precedenti i soggetti che si sottoponevano a task caratterizzati da un feedback audio legato all'errore di posizione del paziente rispetto al target di riferimento (corrispettivo di APER in Variable Amplitude) mostravano performance simili nel ripetere il test senza feedback audio (corrispettivo di NF in Variable Amplitude) contrariamente a ciò che accade ora. Nel test

Variable Amplitude i soggetti sottoposti a feedback audio relativo all'errore della loro posizione rispetto al target di riferimento danno risultati peggiori rispetto quelli a cui non viene dato feedback audio. Probabilmente la spiegazione sta nel fatto che i test precedenti non richiedevano adaptation: è possibile che per il test Variable Amplitude l'aggiornamento del modello interno è reso maggiormente difficoltoso dall'eccesso di informazione data dal feedback di posizione.

- nei test precedenti i soggetti sottoposti al feedback audio legato alla variazione della velocità del target di riferimento (corrispettivo di ATRV in Variable Amplitude) avevano performance migliori rispetto ai soggetti che non ricevevano feedback audio (corrispettivo di NF in Variable Amplitude) come anche nel caso del test Variable Amplitude. L'adattamento del modello interno diventa più semplice in presenza di una descrizione maggiore dell'ambiente.

4.4.4 Confronto tra lead error e lag error

Nei test effettuati di fondamentale importanza risulta l'analisi legata al ritardo e/o all'anticipo dei movimenti del soggetto in esame rispetto il target di riferimento. Nei grafici sottostanti sono stati affiancati i rispettivi risultati ottenuti per il *lead error* ed il *lag error*. Una prima osservazione ci permette di osservare che



(a) Grafico riferito al 'lead error' (b) Grafico riferito al 'lag error'

Figura 4.6: Confronto tra lead e lag error

il gruppo 2 (ovvero quello formato dai soggetti sottoposti ad un feedback audio relativo all'errore di posizione) sembrano essere portati ad anticipare troppo il

movimento relativo all'esercizio da svolgere rispetto ai soggetti appartenenti ad altri gruppi. Infatti i gruppi NF, ATRV e ATRJ risultano essere in ritardo rispetto al target di riferimento. Dal punto di vista dell'analisi statistica osserviamo come i rispettivi confronti fra il gruppo APER e i due gruppi ATRV e ATRJ siano da considerarsi nettamente rilevanti, quindi un'applicazione al soggetto di feedback audio legato all'errore di posizione causa una performance peggiore evidenziata dai valori medi di ritardo dal target e di anticipo sul target. Meno evidenti risultano essere i confronti tra il gruppo APER e il gruppo NF che presentano pValue comunque prossimi alla soglia analizzata. Un'ultima conclusione riguarda la complementarità fra lead error e lag error che balza subito all'occhio osservando i grafici.

4.4.5 Errore medio (emy) ed errore quadratico medio in y (eqmy)

Per la modalità di esercizio di target tracking che prevedeva l'inseguimento a video di un target che si muoveva lungo l'asse x, mediana del display, risulta ovviamente punto focale trattare nel dettaglio l'errore relativo al ritardo in x del soggetto. Ma per uno studio comunque ad ampio raggio della cosa si è scelto

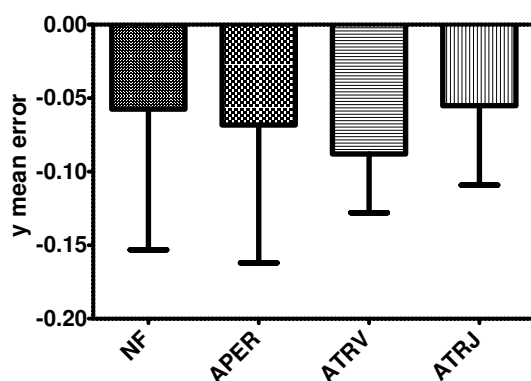


Figura 4.7: Grafico riferito al ritardo medio in y nella fase Var_Test di Variable Amplitude

di analizzare anche i relativi campi d'errore legati al moto del soggetto sull'asse

y. Nell'immagine soprastante possiamo notare come nonostante le tipologie di feedback audio applicate ai gruppi fossero diverse, l'errore relativo al ritardo in y risulta equiparabile tra i gruppi stessi. Questo sembra voglia dire che emy non è condizionato dalla tipologia di feedback audio applicato ma non vi è nessuna evidenza statistica nel supportare questo. Per tutti e quattro i gruppi notiamo come tendenzialmente i soggetti sottoposti al test sono portati a rimanere "sotto" la traiettoria di riferimento del target (valore dell'errore <0). Nonostante non presenti informazioni di spicco presentiamo nella figura sottostante il grafico relativo all'errore quadratico medio in y .

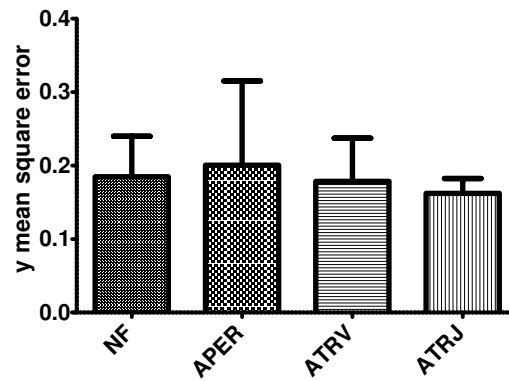


Figura 4.8: Grafico riferito all'errore quadratico medio in y nella fase Var_Test di Variable Amplitude

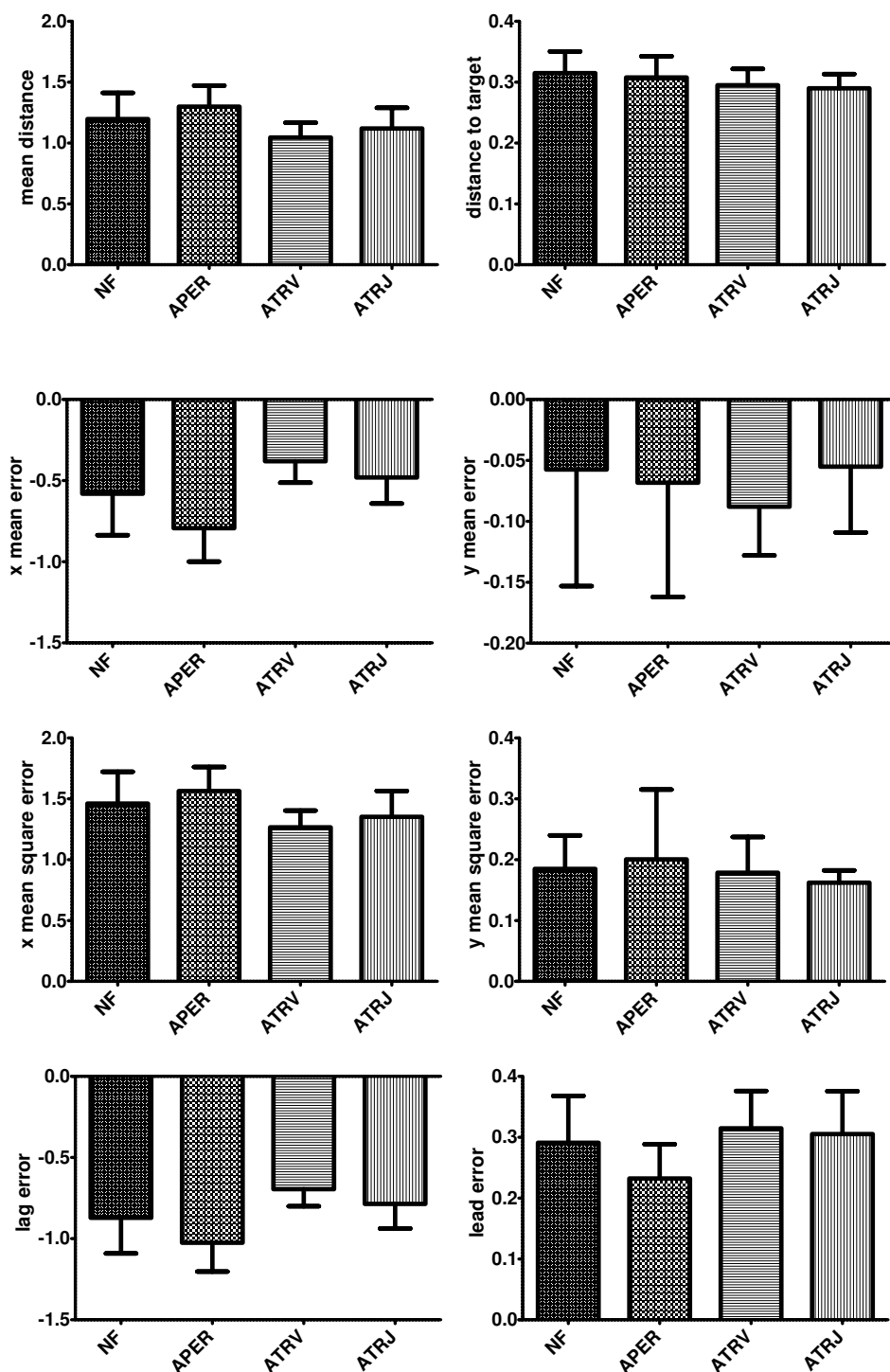


Figura 4.9: Grafici relativi ai parametri considerati nell'elaborazione: distm, dtt, emx, emy, eqmx, eqmy, lag, lead. Tutte le grandezze sono state normalizzate rispetto al raggio del target tranne il dtt, normalizzato rispetto alla lunghezza del tratto percorso.

P-VALUE relativi al Var-Test (Mann Whitney)

	NF-APER	NF-ATRV	NF-ATRJ	APER-ATRV	APER-ATRJ	ATRV-ATRJ
var-distm	0.2071	0.1106	0.6985	*0.0025	* 0.0161	0.2428
var-dtt	0.6891	0.2242	0.1131	0.4996	0.2225	0.7802
var-emx	*0.0392	0.0806	0.5495	*0.0003	*0.0017	0.1823
var-emy	0.8294	0.2545	0.8603	0.0597	0.9212	0.1823
var-eqmx	0.2549	0.0683	0.4597	*0.0050	*0.0161	0.2428
var-eqmy	0.8777	0.8792	0.3787	0.9151	0.8691	0.6607
var-lag	0.0605	0.0683	0.7513	*0.0006	*0.0014	0.0947
var-lead	*0.0289	0.5433	0.6472	*0.0062	*0.0134	0.7802

P-VALUE var (Unpaired t-test)

	NF-APER	NF-ATRV	NF-ATRJ	APER-ATRV	APER-ATRJ	ATRV-ATRJ
var-distm	0.2230	0.0777	0.3759	*0.0013	*0.0239	0.2943
var-dtt	0.6342	0.1842	0.0764	0.3712	0.1828	0.6687
var-emx	*0.0387	*0.0485	0.3042	*<0.0001	*0.0009	0.1587
var-emy	0.7870	0.3838	0.9474	0.5627	0.7001	0.1539
var-eqmx	0.2958	0.0594	0.3126	*0.0011	*0.0255	0.3157
var-eqmy	0.6830	0.8093	0.2357	0.6059	0.3098	0.4170
var-lag	0.0772	*0.0411	0.3175	*<0.0001	*0.0031	0.1519
var-lead	*0.0481	0.4634	0.6635	*0.0047	*0.0137	0.7602

P-VALUE var (Unpaired t-test with Welch's)

	NF-APER	NF-ATRV	NF-ATRJ	APER-ATRV	APER-ATRJ	ATRV-ATRJ
var-distm	0.2289	0.0656	0.3710	*0.0009	*0.0242	0.2868
var-dtt	0.6348	0.1734	0.0730	0.3542	0.1688	0.6718
var-emx	*0.0417	*0.0399	0.2955	*<0.0001	*0.0008	0.1557
var-emy	0.7873	0.3540	0.9462	0.5232	0.6867	0.1488
var-eqmx	0.3036	*0.0496	0.3086	*0.0007	*0.0275	0.3069
var-eqmy	0.6758	0.8109	0.2273	0.5744	0.2786	0.4465
var-lag	0.0814	*0.0339	0.3106	*<0.0001	*0.0029	0.1457
var-lead	0.0531	0.4535	0.6623	*0.0061	*0.0170	0.7587

Tabella 4.1: Tabelle p-value calcolati rispettivamente con Mann-Whitney, Unpaired t-test e Unpaired t-test con Welch's correction. Sono evidenziati con un asterisco i valori di p-value<0,05

4.4.6 Riassunto dei risultati legati al Var_Test

In base alle considerazioni fatte in quest'ultimo capitolo è possibile esprimere le conclusioni di questo lavoro sviluppandole in alcuni punti fondamentali:

- Qualunque gruppo si voglia prendere in considerazione, l'esercizio di target tracking proposto risultava essere complesso da eseguire. A qualunque tipologia di feedback audio fossero sottoposti i soggetti i valori legati alla distanza dal target erano comunque elevati.
- I soggetti appartenenti al gruppo a cui veniva sottoposto il feedback audio legato all'errore di posizione rispetto al target di riferimento presentano, in termini di accuratezza, esiti meno soddisfacenti rispetto ai soggetti che rispettivamente appartenevano ai gruppi privi di audio o con audio proporzionale alla velocità del target (sia con riferimento alla scala video che scala joystick)
- Il gruppo caratterizzato dal feedback audio legato alla velocità del target con scalatura a video e scalatura rispetto al joystick, risultano essere dal punto di vista dell'accuratezza di un più valido aiuto rispetto all'audio legato all'errore di posizione.
- In termini di performance il gruppo di soggetti caratterizzati da feedback audio legato all'errore di posizione rispetto al target, risultavano essere penalizzati rispetto agli altri gruppi. Si basti vedere l'errore medio rispetto alla direzione del moto. Sembra infatti che il feedback audio in questione non apporti nessuna agevolazione ai soggetti ai quali è stato applicato.

Conclusioni

In questo lavoro di tesi è stato sviluppato un programma in Matlab in grado di generare un'interfaccia GUI capace di interfacciarsi con diversi software (Simulink e Pure Data) legati a loro volta a diversi dispositivi di input (joystick) o output (cuffie e monitor). L'interfaccia grafica creata c'ha permesso di testare diversi soggetti su un'ambiente di realtà virtuale opportunamente creato. Il codice creato è frutto di uno sviluppo di lavori precedenti. I risultati ottenuti dalle elaborazioni sono di buon auspicio per un utilizzo efficace del feedback audio in sedute di riabilitazione. Nonostante si siano effettuati i test su dei soggetti sani si è riuscito a riscontrare la presenza di qualche interessante aspetto connesso alla presenza del feedback audio durante l'esecuzione dei task. La tipologia di feedback audio applicata presenta risultati differenti sulle performance dei soggetti considerati, ecco quindi che un determinato tipo di feedback deve essere progettato con criterio ai fini di permettere al soggetto un più facile ed intuitivo apprendimento. Le diverse tipologie di feedback audio influenzano quindi in maniera diversa l'adaptation alla continua variazione di scala, e fra le tipologie di feedback implementati si può confermare che il feedback audio proporzionale alla velocità del target risulti quello che spinge i soggetti a far meglio. Eventuali proposte future potrebbero prevedere nuove tipologie di feedback audio da applicare a tale test (come ad esempio l'audio proporzionale all'errore di velocità) oppure proporre tipologie di traiettorie differenti, e in aggiunta lo studio di ulteriori parametri d'errore per descrivere in maniera più esaustiva il livello di prestazione raggiunto nel test.

Bibliografia

- [1] G. Rosati, “The place of robotics in post-stroke rehabilitation,” *Expert Review of Medical Devices*, vol. 7, no. 6, pp. 753–758, 2010.
- [2] P. L. Kolominsky-Rabas *et al.*, “Lifetime cost of ischemic stroke in Germany: Results and national projections from a population-based stroke registry,” *Stroke*, vol. 37, pp. 1179–1183, 2006.
- [3] W. S. Harwin, J. L. Patton, and V. R. Edgerton, “Challenges and opportunities for robot-mediated neurorehabilitation,” *Proceedings of the IEEE, Special issue on medical robotics*, vol. 94, no. 9, pp. 1717–1726, 2006.
- [4] A. A. Timmermans, H. A. M. Seelen, R. D. Willmann, and H. Kingma, “Technology-assisted training of arm-hand skills in stroke: concepts on reacquisition of motor control and therapist guidelines for rehabilitation technology design,” *Journal of NeuroEngineering and Rehabilitation*, vol. 6, no. 1, 2009.
- [5] S. Masiero, E. Carraro, C. Ferraro, P. Gallina, A. Rossi, and G. Rosati, “Upper limb rehabilitation robotics after stroke: a perspective from the University of Padua, Italy,” *Journal of Rehabilitation Medicine*, vol. 41, no. 12, pp. 981–985, 2009.
- [6] R. Secoli, G. Rosati, and D. J. Reinkensmeyer, “Using sound feedback to counteract visual distractor during robot-assisted movement training,” in *Proceedings of the IEEE 8th International Workshop on Haptic Audio-Visual Environments and Games HAVE2009*, Lecco, Italy, November 7-8 2009.

-
- [7] G. Rosati, A. Rodà, F. Avanzini, and S. Masiero, “On the role of auditory feedback in robot-assisted movement training after stroke,” 2011, submitted for publication.
- [8] R. Secoli, M.-H. Milot, G. Rosati, and D. J. Reinkensmeyer, “Effect of visual distraction and auditory feedback on patient effort during robot-assisted movement training after stroke,” 2011, under review.
- [9] G. Rosati, F. Oscari, S. Spagnol, and F. Avanzini, “Using sound feedback to improve performance during movement training: can task-related feedback be more efficacious than sound rendering of kinematic error?” 2011, submitted for publication.
- [10] H. S. Jorgensen *et al.*, “Acute stroke with atrial fibrillation: The copenhagen stroke study,” *Stroke*, vol. 27, pp. 1765–1769, 1996.
- [11] C. Bosecker, L. Dipietro, B. T. Volpe, and H. I. Krebs, “Kinematic robot-based evaluation scales and clinical counterparts to measure upper limb motor performance in patients with chronic stroke,” *Neurorehabilitation and Neural Repair*, vol. 24, no. 1, pp. 62–69, 2010.
- [12] G. Rosati, P. Gallina, and S. Masiero, “Design, implementation and clinical tests of a wire-based robot for neurorehabilitation,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 15, no. 4, pp. 560–569, 2007.
- [13] B. Kollen, G. Kwakkel, and E. Lindeman, “Functional recovery after stroke: A review of current developments in stroke rehabilitation research,” *Reviews on Recent Clinical Trials*, 2006, 1, 75-80, vol. 1, pp. 75–80, 2006.
- [14] L. Richards, C. Hanson, M. Wellborn, and A. Sethi, “Driving motor recovery after stroke,” *Topics in Stroke Rehabilitation*, vol. 15, no. 5, pp. 397–411, 2008.
- [15] P. Langhorne, F. Coupar, and A. Pollock, “Motor recovery after stroke: a systematic review,” *The Lancet Neurology*, vol. 8, pp. 741–754, 2009.

-
- [16] R. Wiles, A. Ashburn, S. Payne, and C. Murphy, "Discharge from physiotherapy following stroke: the management of disappointment," *Social Science and Medicine*, vol. 59, no. 6, pp. 1263–1273, 2004.
- [17] M. J. Johnson, "Recent trends in robot-assisted therapy environments to improve real-life functional performance after stroke," *Journal of NeuroEngineering and Rehabilitation*, vol. 3, no. 29, 2006.
- [18] G. B. Prange *et al.*, "Systematic review of the effect of robot-aided therapy on recovery of the hemiparetic arm after stroke," *Journal of Rehabilitation Research and Development*, vol. 43, no. 2, pp. 171–184, 2006.
- [19] G. Rosati, J. E. Bobrow, and D. J. Reinkensmeyer, "Compliant control of post-stroke rehabilitation robots: using movement-specific models to improve controller performance," in *Proceedings of the ASME International Mechanical Engineering Congress & Exposition IMECE 2008*, Boston, MA, USA, Oct 31 - Nov 6 2008.
- [20] P. M. Van Vliet and G. Wulf, "Extrinsic feedback for motor learning after stroke: What is the evidence?" *Disability and Rehabilitation*, vol. 28, pp. 831–840, 2006.
- [21] R. J. Nudo, "Postinfarct cortical plasticity and behavioral recovery," *Stroke*, vol. 38, no. 2, pp. 840–845, 2007.
- [22] J. H. Carr and R. B. Shepherd, *Stroke Rehabilitation - Guidelines for Exercise and Training to Optimize Motor Skill*. Butterworth-Heinemann, 2003.
- [23] World Health Organization, *The World health report 2008: Primary Health Care, Now More Than Ever*. WHO Press, Geneva, Switzerland, 2008.
- [24] ———, *World health statistics 2008*. WHO Press, Geneva, Switzerland, 2008.
- [25] L. Dipietro, H. I. Krebs, S. E. Fasoli, B. T. Volpe, J. Stein, C. Bever, and N. Hogan, "Changing motor synergies in chronic stroke," *Journal of Neurophysiology*, vol. 98, p. 12, August 2007.

- [26] T. Truelsen, B. Piechowski-Jozwiak, R. Bonita, C. Mathers, J. Bogousslavsky, and G. Boysen, "Stroke incidence and prevalence in Europe: a review of available data," *European Journal of Neurology*, vol. 13, no. 6, pp. 581–598, 2006.
- [27] D. J. Reinkensmeyer, J. A. Galvez, L. Marchal, E. T. Wolbrecht, and J. E. Bobrow, "Some key problems for robot-assisted movement therapy research: a perspective from the University of California," in *Proceedings of the IEEE 10th Int. Conf. on Rehabilitation Robotics ICORR2007*, Noordwijk, The Netherlands, June 2007, pp. 1009–1015.
- [28] T. G. Hornby, D. D. Campbell, J. H. Kahn, T. Demott, J. L. Moore, and H. R. Roth, "Enhanced gait-related improvements after therapist- versus robotic-assisted locomotor training in subjects with chronic stroke: A randomized controlled study," *Stroke*, vol. 39, pp. 1786–1792, 2008.
- [29] E. T. Wolbrecht, V. Chan, D. J. Reinkensmeyer, and J. E. Bobrow, "Optimizing compliant, model-based robotic assistance to promote neurorehabilitation," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 16, no. 3, pp. 286–297, 2008.
- [30] J. Hidler, D. Nichols, M. Pelliccio, K. Brady, D. D. Campbell, J. H. Kahn, and T. G. Hornby, "Multicenter randomized clinical trial evaluating the effectiveness of the Lokomat in subacute stroke," *Neurorehabilitation and Neural Repair*, vol. 23, no. 1, pp. 5–13, 2009.
- [31] T. N. Taylor *et al.*, "Lifetime cost of stroke in the united states," *Stroke*, vol. 27, pp. 1459–1466, 1996.
- [32] D. Ward, A. Drahota, D. Gal, M. Severs, and T. P. Dean, "Care home versus hospital and own home environments for rehabilitation of older people," *Cochrane Database of Systematic Reviews*, no. 4, 2008.
- [33] Stroke Unit Trialists' Collaboration, "Organised inpatient (stroke unit) care for stroke," *Cochrane Database of Systematic Reviews*, no. 4, 2007.

- [34] Outpatient Service Trialists, “Therapy-based rehabilitation services for stroke patients at home,” *Cochrane Database of Systematic Reviews*, no. 1, 2003.
- [35] V. S. Huang and J. W. Krakauer, “Robotic neurorehabilitation: a computational motor learning perspective,” *Journal of NeuroEngineering and Rehabilitation*, vol. 6, no. 5, 2009.
- [36] H. I. Krebs, B. Volpe, and N. Hogan, “A working model of stroke recovery from rehabilitation robotics practitioners,” *Journal of NeuroEngineering and Rehabilitation*, vol. 6, no. 6, 2009.
- [37] D. Lloyd-Jones *et al.*, “Heart disease and stroke statistics-2009 update: A report from the American Heart Association Statistics Committee and Stroke Statistics Subcommittee,” *Circulation*, vol. 119, pp. e21–e181, 2009.
- [38] J. Mehrholz, T. Platz, J. Kugler, and M. Pohl, “Electromechanical and robot-assisted arm training for improving arm function and activities of daily living after stroke,” *Cochrane Database of Systematic Reviews*, no. 4, 2008.
- [39] J. Mehrholz, C. Werner, J. Kugler, and M. Pohl, “Electromechanical-assisted training for walking after stroke,” *Cochrane Database of Systematic Reviews*, no. 4, 2007.
- [40] G. Kwakkel, B. J. Kollen, and H. I. Krebs, “Effects of robot-assisted therapy on upper limb recovery after stroke: A systematic review,” *Neurorehabilitation and Neural Repair*, vol. 22, pp. 111–121, 2007.
- [41] K. Mononen, “The effects of augmented feedback on motor skill learning in shooting,” 2007.
- [42] R.A. Magill, “Motor learning and control: Concepts and applications,” *McGraw-Hill*, 2003.
- [43] P.M.V. Vliet, “Extrinsic feedback for motor learning after stroke: What is the evidence,” *Disability and Rehabilitation*, vol. 28, 2006.

-
- [44] B.A.Shabbott and R.L.Sainburg, “Learning a visuomotor rotation: simultaneous visual and proprioceptive information is crucial for visuomotor remapping,” *Springer*, p. 13, March 2010.
- [45] J.A.Taylor and K.A.Thoroughman, “Divided attention impairs human motor adaptation but not feedback control,” *Journal of Neurophysiology*, p. 10, April 2007.
- [46] J.VG.Robertson, T.Hoellinger, P.Lindberg, D.Bensmail, S.Hanneton, and A.Roby-Brami, “Effect of auditory feedback differs according to side of hemiparesis:a comparative pilot study,” *Journal of NeuroEngineering and Rehabilitation*, p. 11, December 2009.
- [47] A.Fishbach and F.A.Mussa-Ivaldi, “Seeing vs. believing: conflicting immediate and predicted feedback lead to sub-optimal motor performance,” *Journal of Neurophysiology*, p. 18, December 2008.

Ringraziamenti

Un ringraziamento sentito al Prof. Rosati Giulio e al dottorando Oscari Fabio perchè oltre al ruolo che hanno rivestito in questo lavoro/progetto di tesi hanno dedicato moltissimo del loro tempo per poterci seguire al meglio nonostante i tempi siano stati sempre limitati. Ringrazio mio padre e mia sorella che in tutti questi anni di studio mi hanno sempre sostenuto e appoggiato sotto tutti i punti di vista, sia morali che economici. Ringrazio tutte le persone che mi sono state vicine in questi anni di studio, che mi hanno permesso di vivere giorno per giorno momenti di grande gioia e di condivisione. Ringrazio Chiara per aver portato tanta, ma tanta pazienza con me. Ringrazio il mio amico Goffo, come lo chiamo sempre io, per avermi da sempre regalato l'amicizia che non ho mai dovuto cercare perchè ho sempre trovato in lui. Ringrazio i miei compagni di studio più intimi Ciak, Mattia, Ida, Lea, Valentino, Maschio e tutti coloro che lo sono stati: Spola, Nucci, Scanf, Giulio, Andrea, Luca, Monica e Alice. Ringrazio gli amici del mio paese che anche se con ritmi di vita differenti hanno sempre condiviso con me i momenti divertenti della mia vita. Ringrazio tutti i ragazzi che ho conosciuto all'università con i quali non si è mai instaurato un rapporto tra colleghi, ma un vero rapporto di amicizia. Con una sola frase vorrei dire a tutti coloro che mi sono vicini che nonostante siano poche le volte in cui è possibile ritrovarci tutti assieme, io vi ho e vi porterò sempre nel mio cuore perchè voi siete parte della forza che mi permette di andare avanti. Il lavoro da me presentato lo dedico però ad una persona che regna dentro di me che avrebbe desiderato tanto vedermi laureato ma solo ora riesco ad esaudire questo mio grande desiderio che in parte era anche il suo: mi sono laureato mamma!