

UNIVERSITÀ DEGLI STUDI DI PADOVA
FACOLTÀ DI INGEGNERIA



Finito di scrivere il giorno 26 marzo 2011 utilizzando L^AT_EX 2_ε

UNIVERSITÀ DEGLI STUDI DI PADOVA
FACOLTÀ DI INGEGNERIA

—
DIPARTIMENTO DI INNOVAZIONE MECCANICA E GESTIONALE
—

TESI LAUREA TRIENNALE IN INGEGNERIA BIOMEDICA

STUDIO DEL RUOLO DEL
FEEDBACK SONORO NEL
CONTROLLO MOTORIO E
NELL'APPRENDIMENTO DI TASK
MOTORI

RELATORE: CH.MO PROF. ING. GIULIO ROSATI

LAUREANDO: ALICE MARCATO

ANNO ACCADEMICO 2010-2011

a mio padre, mia madre e mia sorella...

“ Cento volte al giorno mi capita di pensare che la mia vita, quella interiore e quella pubblica, dipende dal lavoro di altre persone, vive e scompare e mi dico che devo impegnarmi a restituire come e quanto ho ricevuto e ancora ricevo. ”

ALBERT EINSTEIN

Indice

Sommario	XIII
Introduzione	XV
1 Robotica in riabilitazione	1
1.1 Il ruolo della robotica riabilitativa	1
1.1.1 La potenzialità dei sistemi robotici in riabilitazione	3
1.1.2 Efficacia clinica dell'approccio robotico	5
1.1.3 Prospettiva di medio periodo	8
1.2 Motor Control e Motor Learning	11
1.3 Feedback	16
1.3.1 Visione e propriocezione	18
1.3.2 Feedback audio	19
1.4 Obiettivi	21
2 Sistema sviluppato	25
2.1 Architettura del sistema	25
2.2 Joystick a due gradi di libertà	27
2.2.1 Modello Simulink del joystick	29
2.3 Utilizzo dell'interfaccia GUI	32
2.4 Implementazione dell'interfaccia GUI	39
2.4.1 Fattore di scalatura	41
2.4.2 Aggiornamento delle variabili: Funzione Draw	42
2.5 Feedback audio	48
2.5.1 Regole di creazione dei pacchetti - protocollo OSC	48

2.5.2	Pacchetti creati dall'interfaccia	50
2.5.3	Invio dei pacchetti	51
2.5.4	Software di generazione di suoni: Pure Data	52
2.5.5	Modifiche apportate ai suoni scelti	53
2.6	Feedback visivo	54
3	Test Sperimentali	57
3.1	Test precedenti	57
3.2	Definizione del test	60
3.2.1	Fasi del Test	60
3.2.2	Tipologie Feedback Audio	68
3.3	Popolazione	70
3.3.1	Questionario	71
3.3.2	Gruppi ottenuti mediante randomizzazione	72
3.4	Esecuzione dei test	73
3.4.1	Problemi riscontrati	74
4	Analisi statistica	81
4.1	Parametri calcolati per l'elaborazione	81
4.1.1	Distanza media dal target (dism)	82
4.1.2	Integrale del modulo della velocità relativa (dtt)	82
4.1.3	Ritardo medio (emx)	83
4.1.4	Errore medio di posizione in y (emy)	85
4.1.5	Errore quadratico medio in x (eqmx)	85
4.1.6	Errore quadratico medio in y (eqmy)	85
4.1.7	Lead error (lead)	86
4.1.8	Lag error (lag)	86
4.1.9	Rox (ratio on x)	86
4.1.10	τ – costante di tempo	87
4.2	Test per l'analisi statistica	89
4.2.1	Test per dati non appaiati	89
4.2.2	Test per dati appaiati	91
4.3	Elaborazioni dei dati	94

4.3.1	GraphPad Prism	95
4.3.2	Procedura	96
4.4	Risultati Variabile	98
4.5	Risultati Pre_Test	102
4.5.1	Pre-Direct	102
4.5.2	Pre-Pin	102
4.5.3	Pre-PFin	103
4.5.4	Pre-Deltap	110
4.5.5	Pre-Deltap-App	110
4.5.6	Pre-Tau	116
4.6	Risultati Post_Test	118
4.6.1	Post-After	118
4.6.2	Post-Pin	118
4.6.3	Post-PFin	119
4.6.4	Post-Deltap	127
4.6.5	Post-Deltap-App	127
4.6.6	Post-Tau	132
4.7	Confronto Pre/Post	134
4.7.1	Learning	134
4.7.2	Retention	134
	Conclusioni	141
	Bibliografia	143

Sommario

Tra gli effetti dell'ictus cerebrale, dopo la fase di stabilizzazione, si denota un'insorgenza di deficit a livello motorio. 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 in questa 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. Per realizzare tale obiettivo 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 sono i gruppi nei quali sono stati suddivisi i diversi soggetti. Ogni soggetto è stato sottoposto al medesimo esercizio di target tracking quindi ciascun gruppo differisce dall'altro non in base al tipo di esercizio che è stato svolto (infatti era il medesimo per tutti), bensì per la tipologia di feedback audio al quale i soggetti appartenenti a quel gruppo sono stati sottoposti. Per ciascun soggetto è stata valutata l'accuratezza della sua performance mediante il calcolo di numerosi 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]. 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[2, 3]. Finora sono stati proposti molti protocolli terapeutici e soluzioni tecniche, che 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 [3]. 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 [3, 4].

La conoscenza su come operi il motor learning durante la neuro-riabilitazione non è però ancora sufficiente per imporre il design del dispositivo di terapia robotica [5], sebbene siano state proposte recentemente alcune indicazioni in questa direzione [6]. 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 [7].

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 [7] [8] [9] [10], 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 le modalità audio in particolari situazioni video. 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 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 raccolti 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 determinate informazioni. In particolare, verranno con-

frontate differenti tipologie di feedback audio: lo scopo è quello di ottenere una maggior conoscenza riguardo alla modalità con cui tali effetti sonori intervengono nell'esecuzione dei task motori, che prevedono l'utilizzo di particolari modalità video.

Capitolo 1

Robotica in riabilitazione

In questo capitolo vengono presentati i risultati della ricerca bibliografica, realizzata utilizzando motori di ricerca quali *PudMed*, *EngineeringVillage* e *Google*. La ricerca spazia tra articoli degli ultimi dieci anni ed è stata ampliata a studi di anni precedenti per gli articoli di maggiore interesse, grazie ai riferimenti bibliografici presenti. 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 proposte due tabelle che riassumono il contenuto della revisione della letteratura effettuata. La prima si riferisce ai soli articoli citati in questo contesto, la seconda 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 primari della società nei paesi sviluppati, a causa dell'invecchiamento della popolazione e per l'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 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[2, 3]. Finora sono stati proposti molti protocolli terapeutici e soluzioni tecniche, che 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 [3]. 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 [3, 4].

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 [11, 12]. In secondo luogo, il numero di persone che necessitano di riabilitazione dopo l'ictus è in rapida crescita a causa dell'invecchiamento della popolazione [13]. 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; 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 [14]. 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% [15]. Percentuali simili sono riportate da un recente studio dell' Erlangen Stroke Registry in Germania [16]. 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 [14, 17, 18]. L'obiettivo principale della riabilitazione per un soggetto affetto 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' [19]. 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 (< 6 mesi) [4] e in una struttura specializzata [20]. Tuttavia, i programmi tradizionali di 'training' dipendono molto dalla formazione del fisioterapista e dalla sua esperienza [21], e, talvolta, forniscono una quantità limitata di trattamenti per soggetto a causa della mancanza di terapisti [12], che è il problema crescente. Per di più, la valutazione della mobilità del paziente e delle sue capacità funzionali è fatta in gran parte da operatori sanitari basandosi su scale di valutazione standardizzate e approvate, ma pur sempre soggettive [22]. Questi fattori rendono difficile il confronto tra i diversi protocolli di trattamento.

1.1.1 La potenzialità dei sistemi robotici in riabilitazione

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 [2]. 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' [3]. 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...) [3]. 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 [22]. L'accettazione della tecnologia robotica 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 [25].

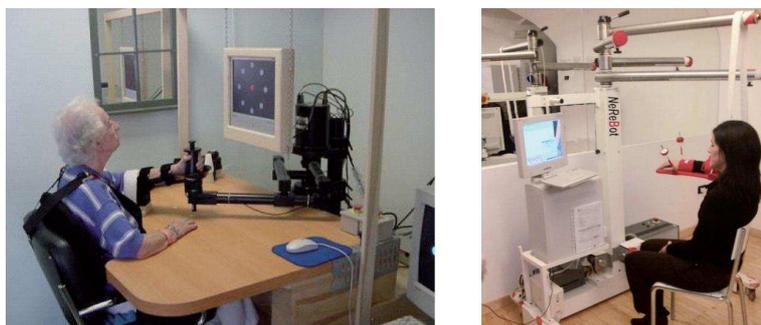


Figura 1.1: Due esempi di dispositivi robotici per la riabilitazione: MIT-Manus e NeReBot

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 [3]. D'altra parte, la riduzione del contatto fisico con una persona può diventare uno svantaggio [12], 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.

1.1.2 Efficacia clinica dell'approccio robotico

Nonostante i potenziali benefici dei movimenti di training mediati da robot nel post-ictus [2], 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 [26, 27]. 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, persino 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 [12] (come un terapeuta 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 [27] 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 [28], 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 subacuta che in fase cronica [29, 30]. 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 che non sono 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 movi-

menti 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 [3]. 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 [31] 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.



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

1.1.3 Prospettiva di medio periodo

Grazie ai risultati positivi ottenuti fin 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 [3]. 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 [12, 22]).

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 stimolante 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 neuro-degenerative. Per esempio, sono stati presentati recentemente i risultati di test clinici con dispositivi robotici sulla sclerosi multipla e su pazienti pediatrici con lesioni corticali[4].

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 signi-

ficativamente il risultato funzionale, durante il ricovero ospedaliero del paziente [2, 4]. 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 [3, 4, 32, 33].

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 [32, 34]. 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 [35].

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 [5], sebbene siano state proposte recentemente alcune indicazioni in questa direzione [6]. 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 [7]. 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 [13]. 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 [36, 37]. 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 [2, 38]. 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 [39]. 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 [2], 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 Motor Control e Motor Learning

Per controllo motorio (*motor control*) si intende la capacità del sistema nervoso di regolare o dirigere il movimento. La regolazione dell'attività motoria volontaria, quale sia il suo livello di complessità, è effettuata dalla corteccia motoria primaria, situata nella parte posteriore del lobo frontale; la corteccia di ciascun lato dell'encefalo controlla principalmente i muscoli sul lato opposto del corpo. I tratti nervosi che originano nella corteccia motoria dell'emisfero sinistro si incrociano per poi scendere nel midollo spinale e terminare sui motoneuroni efferenti, i quali determinano la contrazione dei muscoli scheletrici sul lato destro del corpo. La stimolazione di differenti aree della corteccia motoria primaria causa il movimento in differenti regioni del corpo, secondo la rappresentazione distorta delle varie parti del corpo che l'encefalo conserva (nel cosiddetto 'l'homunculus motorio', le dita delle mani o i muscoli della fonazione sono grossolanamente esagerati se confrontati con tronco, braccia e arti inferiori, indicando il livello fine di controllo motorio di queste parti del corpo).

Oltre la corteccia motoria, altre regioni dell'encefalo sono importanti nel controllo motorio: in particolare il cervelletto contribuisce al mantenimento dell'equilibrio, coordina i movimenti volontari spazializzati ed interviene nella programmazione e nell'inizio dell'attività volontaria, fornendo input alle aree motorie corticali.

Il movimento, inoltre, non può essere studiato senza tenere conto della percezione. La percezione è l'interpretazione a livello conscio del mondo esterno creata dall'encefalo in base ad un pattern di impulsi nervosi inviatigli dagli organi di senso. Percezione ed azione sono essenziali l'una all'altra: per poter controllare la contrazione dei muscoli scheletrici al fine di ottenere il movimento desiderato, il SNC deve essere informato continuamente dei risultati della sua azione, attraverso il feedback sensoriale. In particolare, la proprioccezione, ovvero il senso della posizione del proprio corpo o di una sua parte nello spazio, ha fondamentale importanza nell'esecuzione di qualsiasi movimento.

Il *motor learning* è, invece, l'insieme di quei processi associati all'esercizio o

all'esperienza che portano a cambiamenti relativamente permanenti delle capacità di risposta di un soggetto (Schmidt, 1975). Tale apprendimento evolve nel corso di una serie di ripetizioni di un movimento e richiede la presenza di perturbazioni esterne. Pratica e qualità dello studio o del training influiscono moltissimo sulla velocità di apprendimento e sulla performance finale.

La riorganizzazione cerebrale, che porta all'acquisizione di nuove capacità motorie, è possibile grazie all'elevato grado di plasticità dell'encefalo, ovvero la capacità che il cervello ha di modificarsi o rimodellarsi funzionalmente in risposta alle richieste che gli vengono poste. Tale abilità è più pronunciata nei primi anni di sviluppo, ma anche l'encefalo adulto conserva una certa plasticità. In particolare, quando un'area cerebrale associata ad una particolare attività viene distrutta, altre aree dell'encefalo possono assumere gradualmente una parte o la totalità delle funzioni della regione danneggiata. L'apprendimento di nuove abilità motorie con un CNS intatto e il recupero di abilità precedentemente apprese, che sono state perse dopo un danno cerebrale, sembrano essere simili nei vari aspetti. Così, il motor learning è possibile anche quando la corteccia motoria è parzialmente danneggiata, e, seppur più lento, è efficace quanto quello di soggetti sani. Sfruttando questi meccanismi, esercizi post-stroke mirati possono rilanciare la plasticità neurale ed il recupero funzionale dell'arto lesa, evitando un'ulteriore perdita della rappresentazione della mano nella corteccia intatta e inducendo un'espansione del territorio della mano nella corteccia adiacente.

Un concetto teorico fondamentale nello studio di Motor Control e Motor Learning è quello di modello interno, ovvero una rappresentazione inversa della fisica del corpo che il sistema nervoso immagazzina. Tale modello si basa su due differenti meccanismi di controllo:

- meccanismi di controllo feedforward: si basano sulla previsione delle forze che saranno necessarie per l'esecuzione di un determinato movimento, senza l'utilizzo di informazioni sensoriali. Tale meccanismo ha l'evidente inconveniente di non essere in grado di rispondere a perturbazioni inaspettate

- meccanismi di controllo feedback: quando un errore nell'esecuzione del movimento è disponibile tramite le informazioni sensoriali, queste possono essere utilizzate per effettuare correzioni immediate del movimento stesso, in modo che segua il piano desiderato. Tale correzione, tuttavia, ha l'inconveniente di essere disponibile solo dopo un certo ritardo sensoriale

Questi 2 meccanismi lavorano assieme: l'errore di retroazione nel controllore a feedback viene utilizzato come segnale di apprendimento di un modello feedforward.

Tale modello è stato messo in evidenza, per esempio, dallo studio di Mah, Mussa-Ivaldi [40]: la prima parte dell'esperimento (*primary task*) prevede di seguire i movimenti random di un target, bilanciando contemporaneamente un pendolo inverso che si muove su un piano orizzontale sopra un carrello. I soggetti impugnano l'estremità di un manipolatore a 2 gradi di libertà, il quale controlla un software che simula il moto del pendolo. Realistici feedback di forza sono stati forniti per simulare la forza di gravità. I soggetti sono stati in grado di imparare la sequenza corretta di movimenti e di forze necessarie per eseguire il task e di crearsi un modello di tali forze per ripetere l'esercizio il giorno dopo. La seconda parte dello studio (*intervening task*) ripete il primary task con alcune modifiche: i soggetti sono stati divisi in 2 gruppi:

- fast group: il target si muove a velocità doppia con pendolo inverso
- slow group: il target si muove alla stessa velocità con pendolo stabile

Sorprendentemente i soggetti appartenenti allo slow group, che hanno dovuto apprendere nuovi schemi di forze per bilanciare il pendolo, hanno ottenuto i risultati migliori; ciò suggerisce che la ritenzione del task è favorita quando simili movimenti degli oggetti da seguire richiedono risposte motorie incompatibili (teoria *enhanced-response selection*).

Per studiare la capacità adattativa neuro-motoria umana, Mussa-Ivaldi e collaboratori hanno proposto uno schema sperimentale, divenuto lo standard per questi studi. I soggetti, seduti davanti ad un monitor, impugnano l'estremità di

un manipolatore a due gradi di libertà in grado di generare un campo di forze lungo il piano orizzontale; l'esercizio consiste nel raggiungere 3 target, disposti ai vertici di un triangolo equilatero, che in modo random vengono rappresentati sul monitor. Se il robot non applica alcuna forza, le traiettorie appaiono regolari e lineari. L'applicazione del campo di forza, invece, causa inizialmente la curvatura della traiettoria della mano. Con la pratica i soggetti si adattano alle forze imposte e riescono a raggiungere il target previsto. Quando il campo di forza viene improvvisamente rimosso, si osservano i cosiddetti *after-effects*: i soggetti disegnano traiettorie curvate in maniera speculare rispetto a quando è stata generata la perturbazione. Gli autori hanno ipotizzato che l'adattamento causerebbe un'alterazione nel modello interno feedforward, la quale porta il soggetto a predire gli effetti delle sollecitazioni a cui è sottoposto. Ad ogni modo, durante i 50-75 movimenti che seguono la rimozione del campo di forze, le traiettorie tornano lineari (*washed-out*).

Gli studi sul movimento considerano, nella grande maggioranza, azioni di point-to-point reaching, o di tracking di un target su di un piano orizzontale. Questa scelta è dovuta al fatto che tale movimento è uno dei più semplici da eseguire, per l'assenza di forze di gravità nei movimenti e per la facilità di organizzazione sperimentale. Generalmente ai pazienti è chiesto di seguire un target che si muove su un piano orizzontale oppure di raggiungere punti fissi disposti lungo una circonferenza (o lungo i vertici di un triangolo o quadrato). I target vengono opportunamente rappresentati su uno schermo di fronte al soggetto. L'attenta analisi di questi movimenti, seppur semplici, ha portato a definire caratteristiche comuni a svariate altre tipologie di movimento; in particolare:

- la mano si muove verso un target secondo traiettorie approssimativamente rettilinee, indipendentemente dalla direzione e dall'ampiezza del movimento
- il movimento ha un profilo di velocità a campana, con un picco proporzionale alla distanza percorsa. Nei movimenti di reaching, che richiedono un'elevata precisione, tale profilo risulta asimmetrico: infatti, ad una fase iniziale (*transport phase*) durante la quale il braccio è trasportato nelle vicinanze del target, segue una fase di aggiustamento (*correction phase*)

durante la quale vengono eseguite piccole correzioni alle imprecisioni dei movimenti precedenti

- si ipotizza che i movimenti siano formati dalla combinazione di blocchi elementari, chiamati ‘*submovements*’; essi si sovrappongono per formare un movimento fluido, caratterizzato dalla cosiddetta ‘*smoothness*’ (dolcezza). Tale teoria, tuttavia, non ha ancora una dimostrazione univoca, a causa della difficoltà di estrazione dei submovements [41]

I pazienti post-stroke, soprattutto nella fase acuta, non sono in grado di produrre movimenti con tali caratteristiche: in particolare essi presentano curvature delle traiettorie della mano; inoltre, i movimenti appaiono frammentati e stereotipati, presentando un numero eccessivo di submovement e molti picchi di velocità. Non c’è, invece, alcuna differenza, rispetto ad un soggetto sano, nel profilo di velocità di un singolo submovement. Diversi studi [42, 41] hanno evidenziato come il recupero inizi con l’acquisire di nuovo la capacità di creare i submovement e, solamente in un secondo tempo, di combinarli assieme. Durante la terapia i submovements tendono a diminuire in numero e, progressivamente, a sovrapporsi contribuendo all’aumento della smoothness. In particolare, Roher [41] ha esaminato 31 soggetti, 12 in fase acuta (*inpatient*) e 19 in fase cronica (*outpatient*). L’esercizio è standar: eseguire dei movimenti di reaching di 8 target disposti lungo una circonferenza, partendo e ritornando al centro dopo il raggiungimento di ogni target; il MIT-Manus accompagna ogni soggetto durante l’esecuzione del task; si tratta di un robot a 2 gradi di libertà che permette il movimento di spalla e gomito su un piano orizzontale. Tale apparecchiatura è in grado di registrare il comportamento umano ed adattarsi di conseguenza (controllo d’impedenza): la macchina riconosce una componente volontaria del movimento, anche minimale, del paziente e quando questa viene meno si attiva per aiutare il soggetto a terminare il movimento. Il risultato generale è stato un aumento della smoothness, sia per gli inpatient che per gli outpatient. Il calcolo di tale parametro è stato basato sul minimum jerk (terza derivata della posizione rispetto al tempo) e sul numero di picchi di velocità: minore è il jerk e il numero di picchi, più dolce risulta il movimento. Inoltre è stato osservato: (1) un aumento dei picchi di velocità di ogni submovement, (2) una diminuzione del numero di submovement, (3) una diminuzione del tempo che

intercorre tra due picchi di velocità successivi, in particolare per gli inpatient, (4) una progressiva sovrapposizione dei submovements, in particolare per gli inpatient.

Accanto alla scarsa smoothness del movimento, la performance di un paziente colpito da stroke è caratterizzata dalla difficoltà a coordinare i movimenti con più gradi di libertà; in particolare, viene evidenziata una scarsa coordinazione di spalla e gomito, dovuta alla scorretta temporizzazione delle diverse articolazioni. Il soggetto, inoltre, è in grado di muoversi solamente seguendo pochi stereotipati schemi motori: spalla e gomito sono scarsamente indipendenti l'una dall'altro, il che preclude l'esecuzione di molti movimenti in tutte le direzioni. Krebs, Di Pietro [43] hanno studiato un eventuale ripristino della coordinazione di spalla e gomito in 117 pazienti in fase cronica. La prima parte dell'esperimento è del tutto simile a quello condotto da Roher: i soggetti si allenano eseguendo movimenti di reaching di 8 target disposti lungo una circonferenza; anche in questo caso, ogni esecuzione viene assistita dal MIT-Manus. Successivamente viene chiesto di disegnare, più volte, sia in senso orario che antiorario, una circonferenza continua. Tale esercizio viene scelto perché richiede una buona coordinazione di spalla e gomito; in particolare richiede un'ottima temporizzazione tra l'abduzione della spalla e l'estensione del gomito. La bontà dell'esercizio è stata valutata tramite l'*axes ratio* (rapporto tra gli assi), ovvero il rapporto tra l'asse minore e l'asse maggiore dell'ellisse che meglio si adatta al percorso disegnato dalla mano del soggetto. Tale parametro varia tra 0 e 1; maggiore è il valore, maggiore è la qualità del disegno.

Durante il corso della terapia tale parametro ha teso progressivamente a 1, il che indica l'acquisizione da parte dei soggetti del task richiesto.

1.3 Feedback

Uno dei più importanti aspetti del processo di apprendimento, e soprattutto di affinamento, di un compito motorio è l'utilizzo del feedback. Con tale termine ci si riferisce a tutte quelle informazioni che il soggetto riceve durante e dopo l'esecuzione di un task. Una prima classificazione divide il feedback come:

- intrinseco: relativo alle informazioni associate al movimento provenienti dal proprio sistema sensoriale. Il feedback intrinseco fornisce informazioni, sia durante che dopo la produzione di movimento
- estrinseco: è invece relativo alle informazioni provenienti da fonti esterne, come può essere il terapeuta o un dispositivo robotico.

Il feedback estrinseco affianca la risposta che il soggetto ottiene dai propri sensi: visione, udito e propriocezione. Quando il sistema sensoriale non è in grado di fornire informazioni utili alla performance, il feedback estrinseco può rivelarsi risolutivo ai fini dell'apprendimento e dell'esecuzione di un determinato task motorio. In particolare gioca due importanti ruoli: innanzitutto fornisce informazioni riguardo la performance, cioè valuta se l'esercizio ha avuto successo o meno e descrive gli errori che il soggetto dovrebbe correggere per una migliore esecuzione; inoltre aumenta attenzione e motivazione del soggetto, portandolo ad incrementare i propri sforzi per raggiungere l'obiettivo, che è la chiave del recupero funzionale. Nel feedback estrinseco si possono ulteriormente distinguere due tipi di informazione:

- la KR (*knowledge of results*): da un'indicazione che specifica se l'obiettivo dell'esercizio è stato raggiunto o meno. Tali informazioni hanno un maggior effetto ai fini dell'apprendimento quando sono precise, espresse cioè in termini quantitativi, piuttosto che quando sono invece di tipo qualitativo (ad esempio, osservazioni come 'giusto', 'sbagliato'). La precisione non deve però superare il limite di elaborazione del soggetto, per non produrre un degrado della prestazione
- la KP (*knowledge of performance*): da un'indicazione della qualità dei movimenti eseguiti dal paziente per portare a termine un esercizio. Tali informazioni possono essere fornite in vari modi (verbale e non). Anche in questo caso le informazioni di tipo quantitativo sono più efficaci

Diversi autori [44] hanno verificato che i pazienti migliorano la qualità del movimento in un reaching task quando viene fornita loro una conoscenza della performance (KP, istruzioni riguardanti le sequenze di movimenti da compiere) piuttosto che una conoscenza del risultato (KR, precisione del movimento). E' pur vero

che nell'esperimento condotto da Cirstea la KP è stata fornita verbalmente e che il grado di invalidità dei soggetti ha potuto influire sul risultato finale dell'esperimento.

1.3.1 Visione e propriocezione

Al fine di valutare quali possano essere gli stimoli utili da applicare ai pazienti durante la loro riabilitazione, è necessario stabilire quale feedback (visivo, uditivo o di forza) sia più efficace. A tale scopo Feygin [45] ha effettuato un confronto tra tre metodi di allenamento robot-assistito:

- visivo (V), nel quale il soggetto guarda l'end-effector di un robot eseguire il movimento ideale
- aptico (H), dove la visione dell'apparato è impedita, mentre il soggetto impugna l'end-Effector che guida la sua mano lungo il percorso ideale
- aptico-visivo (H+V), nel quale il soggetto guarda il movimento mentre è guidato apticamente

L'allenamento aptico, da solo, si è rivelato meno efficiente dell'allenamento visivo per quanto riguarda la posizione e la forma, ma più efficiente per quanto riguarda la temporizzazione. Per ogni modalità di allenamento, si sono poi studiati gli effetti con due possibilità di richiamo (*recall*): con la visione diretta (H+V) o senza la visione (H). L'effetto del metodo di allenamento è moderato dalla modalità di recall utilizzata; in particolare si osserva un calo di performance dell'allenamento aptico nel caso di recall (H+V), ovvero la visione può in qualche modo interferire con la rappresentazione aptica del movimento.

Un risultato simile è stato ottenuto da Morris [46]: 12 partecipanti vengono guidati passivamente lungo traiettorie disegnate su un piano orizzontale; scopo del test è imparare la corretta sequenza di forze che vengono fornite lungo un piano verticale. Anche questo test prevede 3 differenti condizioni di allenamento:

- aptico (H): il soggetto, immerso in un campo di forze verticale, deve cercare di rimanere sul piano di movimento orizzontale

- visivo (V): l'intensità della forza viene rappresentata lungo un monitor in corrispondenza della posizione del soggetto (il quale rimane vincolato al piano orizzontale di movimento)
- aptico-visivo (H+V): come (V); in aggiunta, viene fornita un'informazione relativa all'intensità della forza applicata in quel momento dal soggetto

Il grado di apprendimento è stato misurato dalla precisione con cui un soggetto riproduce lo schema di forze sentite in un successivo test di richiamo (T). I risultati hanno indicato una maggiore efficacia dell'allenamento (H+V) rispetto all'allenamento visivo o aptico, da solo; inoltre (V) prevale su (H). Tali risultati suggeriscono che l'allenamento aptico, supportato da un adeguato feedback visivo, può rivelarsi uno strumento efficace per l'apprendimento di compiti sensomotori che richiedono la memorizzazione di determinati schemi di forze.

1.3.2 Feedback audio

Mentre la stimolazione visiva è un argomento molto studiato nel campo della riabilitazione post-stroke, ben poca attenzione è dedicata al feedback uditivo. Nonostante ciò, anche il suono può risultare un ottimo strumento ai fini del recupero: stimola, infatti, la corteccia uditiva, posta nel lobo temporale, la quale comunica con la corteccia motoria, integrando le afferenze somatosensoriali del lobo parietale; in tal modo contribuisce a favorire la plasticità del cervello, meccanismo fondamentale per il recupero di danni neurologici, in particolare dell'ictus.

Se propriamente disegnato, il feedback audio può portare molti vantaggi, in quanto:

- stimola l'utente ad eseguire esercizi ripetitivi e stancanti
- può rappresentare informazioni temporali e spaziali che possono migliorare il processo di apprendimento motorio; apre, infatti, un altro canale di comunicazione attraverso il quale vengono fornite informazioni complementari a quelle disponibili tramite la vista; ad esempio, può fornire informazioni difficili da vedere quando lo schermo è limitato
- può sostituire le altre modalità di feedback, in caso di loro assenza, come nel caso di pazienti non vedenti.

Il feedback audio può fare uso o meno del linguaggio; tuttavia, la parola viene preferibilmente sostituita da suoni per fornire un'indicazione quantitativa. Sono stati individuati 4 modi principali in cui i dati possono essere codificati in audio:

- auditory icons: suoni di tutti i giorni riprodotti sul computer che stabiliscono un collegamento intuitivo con ciò che rappresentano
- earcons: toni astratti e sintetici utilizzati per fornire un messaggio; un esempio può essere il 'beep' di inizio o fine esercizio; devono essere spiegati perché non c'è nessun collegamento con quello che rappresentano
- speech: linguaggio parlato
- sonification: 'mappatura di informazioni multidimensionali in un dominio acustico'; può essere considerato l'equivalente della visualizzazione delle informazioni

Fornire un appropriato feedback audio si è rivelato efficace per bilanciare cali di attenzione, dovuti a varie fonti di distrazione, durante l'esecuzione del task [7]. Durante l'esercizio di target tracking, privo di feedback sonoro, veniva introdotto un semplice distrattore visivo, il quale aumentava l'errore di inseguimento (*tracking error*), portando ad un peggioramento della performance. In alcuni test veniva fornito un feedback audio, in forma di beep, proporzionale all'errore di inseguimento, il quale riportava la performance su livelli normali.

Robertson [47] ha studiato l'effetto di 2 tipi di feedback audio forniti a pazienti emiparetici con lesioni all'emisfero destro (*RHD*) o sinistro (*LHD*). L'esercizio prevede movimenti di reaching verso 9 target equamente distribuiti nell'area di lavoro di fronte al soggetto. Il feedback somministrato può essere:

- simple: il volume aumenta quanto più la mano si avvicina al target
- spatial: il suono percepito dipende dalla posizione della mano rispetto al target (è più basso nell'orecchio sinistro se il target è alla destra del soggetto)

Sorprendentemente non sono state trovate differenze tra i 2 tipi di audio. I soggetti, eccetto uno (un musicista), non sono nemmeno stati in grado di riconoscere la natura del feedback spazializzato. Invece, sono comparse differenze tra i

gruppi RHD e LHD. Tale stimolazione si è rilevata positiva per il gruppo RHD, ma ha peggiorato la performance del gruppo LHD; ciò dimostra che l'efficacia del feedback audio dipende dall'emisfero lesionato e che i 2 emisferi giocano un ruolo diverso nel processare le informazioni uditive.

1.4 Obiettivi

Lo scopo di questo lavoro è quello di investigare gli effetti che diversi tipi di feedback portano nell'esecuzione di movimenti di tracking di un target; in particolare si è cercato di capire quali siano i parametri più opportuni ed efficaci a cui legare tale feedback e grande attenzione è stata rivolta al suo ruolo nella performance del soggetto, che può risultare uno strumento molto utile per un migliore e più pronto recupero della capacità motoria; in particolare, si è cercato di approfondire il ruolo che tale feedback gioca nel motor learning e il modo in cui un modello sonoro entra in un modello interno feedforward.

E' stata presa in considerazione una popolazione di 47 soggetti sani, privi di alcun tipo di menomazione motoria. Diversi tipi di feedback audio sono stati sviluppati e forniti online durante l'esecuzione di esercizi di target tracking. Sono stati forniti 3 tipi di feedback, relativi rispettivamente all'errore di posizione del soggetto rispetto al target, alla velocità del target visualizzata a video e alla velocità del target relazionata al movimento del Joystick, mentre ad alcuni soggetti non è stato fornito nessun tipo di effetto sonoro. Tali feedback sono stati somministrati durante l'esecuzione di diversi task, i quali prevedono particolari effetti nella relazione tra visualizzazione video dell'esercizio e il movimento svolto dal soggetto per compierlo; quindi lo scopo è analizzare sia il livello di influenza (positiva o negativa) del feedback audio nella prestazione dei soggetti, ma anche la differenza tra i diversi effetti sonori implementati.

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

Con tale studio si spera di contribuire allo sviluppo del feedback uditivo, il cui potenziale risulta non essere ancora in gran parte sfruttato e la cui integrazione in un dispositivo di riabilitazione può contribuire a superare alcuni dei limiti principali dei sistemi di terapia in corso, sia in termini di coinvolgimento degli utenti che di apprendimento di compiti motori più complessi.

<i>Anno</i>	<i>Autore</i>	<i>Argomento</i>	<i>Device</i>	<i>Soggetti</i>	<i>Feedback</i>
2002	Rohrer	submovemets	Mit-Manus	31s	video
2002	Feygin	motor learning	Phantom 3.0	36s	video,force field
2003	Mussa-Ivaldi	internal model	RIC robot	27h	video, force field
2007	Di Pietro	submovemets	Mit-Manus	117s	video
2007	Cirstea	KR/KP	-	28s, 5h	video, audio
2007	Morris	motor learning	3 dof robot	18s	video, force field
2009	Robertson	feedback audio	-	16s, 10h	video, audio
2009	Reinkensmeyer	feedback audio	Pnew-Wrex	10h	video, audio

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

Capitolo 2

Sistema sviluppato

2.1 Architettura del sistema



Figura 2.1: Postazione di lavoro utilizzata per i test

In figura 2.1 è mostrata la postazione di lavoro che è stata creata per eseguire il test. Il soggetto, seduto alla scrivania, ha di fronte lo schermo del computer sul quale deve eseguire l'esercizio di target tracking. Per comandare il movimento si agisce sul joystick a due gradi di libertà posto sulla destra (sinistra per i manci-

ni). Il sistema di controllo del joystick è stato sviluppato nell'ambiente Simulink di Matlab mentre una scheda di acquisizione, collegata al quadro elettrico e in comunicazione con il computer in LAN, si occupa di gestire l'alimentazione degli azionamenti e leggere la posizione della leva impugnata dal soggetto.

L'operatore, invece, siede di fronte ad un secondo schermo e gestisce le varie fasi dell'esercizio attraverso un'interfaccia GUI, realizzata sempre in Matlab: questa permette la comunicazione tra il modello Simulink, le funzioni che consentono di creare la traiettoria e gli oggetti mobili e il software PureData, deputato alla generazione dei suoni; il feedback video, invece, viene gestito all'interno di Matlab. In figura 2.2 vengono evidenziati i collegamenti logici tra le varie parti del sistema sviluppato.

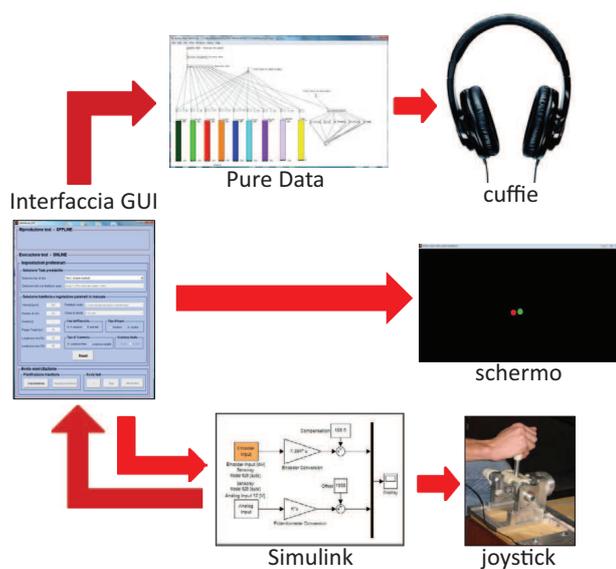


Figura 2.2: Schema che riassume i collegamenti logici tra le varie parti del sistema

2.2 Joystick a due gradi di libertà

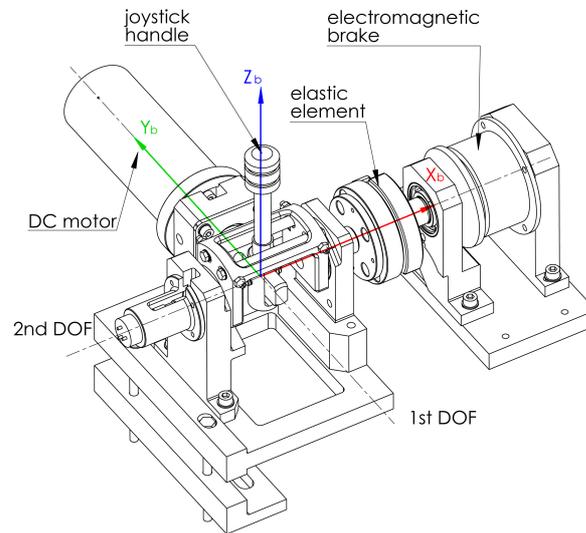


Figura 2.3: Immagine CAD del joystick

Si tratta di una struttura a 2 gradi di libertà, realizzata presso i laboratori del DIMEG; essa comprende un g.d.l. attivo, comandato da un motore elettrico in corrente continua, e un g.d.l. 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 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 movimenti dei soggetti, è stata realizzata una prolunga in plastica. Da ciò risulta che:

- la *lunghezza* totale della leva è 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:

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

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 (figura 2.4).



Figura 2.4: Immagine del joystick utilizzato durante i test

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 Ω e alimentato a 10 V. 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: $2^{16}/2\text{valori} = 32768\text{valori}$), la risoluzione è pari a $360^\circ/32768 = 0,011^\circ$.

2.2.1 Modello Simulink del joystick

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 tempo reale.

La figura 2.5 mostra lo schema del modello del 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:

$$o_0 = ki_0 + v \quad (2.1)$$

$$o_1 = ki_1 + v \quad (2.2)$$

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.

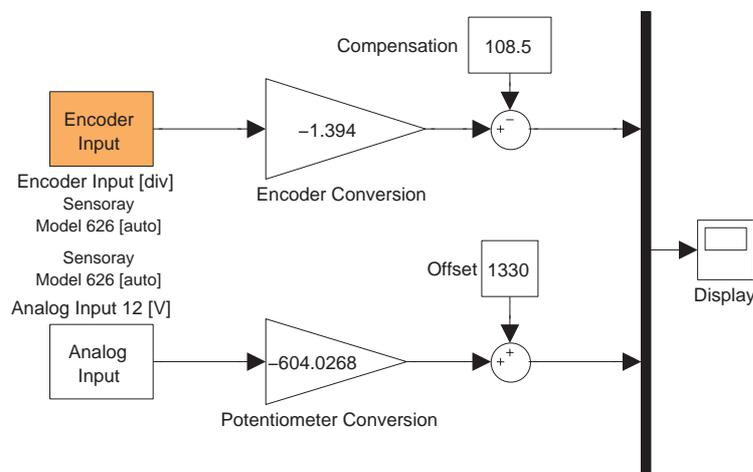


Figura 2.5: Schema del modello Simulink del joystick

È importante notare che lo spostamento della leva lungo i limiti della corsa utile genera un grafico riportato in figura 2.6, 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 2.1 e 2.2 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

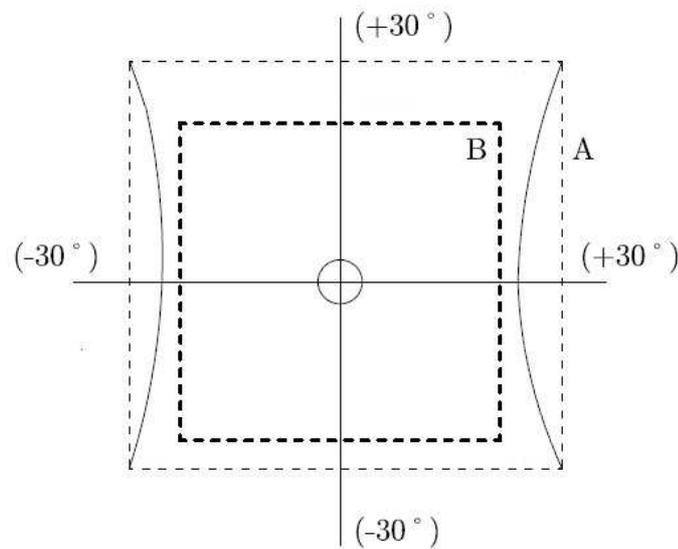


Figura 2.6: Area di lavoro della leva del joystick con evidenziati i limiti individuati tramite la procedura di calibrazione (A = limiti massimi dell'area di lavoro, B = area di lavoro effettiva)

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*, il 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)
% Name
mdl.modelName = modelName;
% Load the simulink model
load_system(mdl.modelName);
% List the blocks that are to have listeners applied
mdl.viewing = struct('blockName','','blockHandle',[],'blockEvent','','blockFcn',
    [],[]);
% Every block has a name
mdl.viewing.blockName = sprintf('s/Display',mdl.modelName);
% That block has a handle
% (This will be used in the graphics drawing callback, and is done here
% as it should speed things up rather than searching for the handle
% during every event callback.)mdl.viewing.blockHandle = get_param(mdl.viewing.
    blockName,'Handle');
mdl.viewing.blockEvent = 'PostOutputs';
% List the block event to be listened for
mdl.viewing.blockFcn = @eventListener;
% List the function to be called
% (These must be subfunctions within this mfile).
mdl.originalStartFcn = get_param(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);
```

2.3 Utilizzo 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* di sviluppo 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, come avviene invece nelle più tradizionali interfacce testuali.

Nel seguente capitolo viene studiato nel dettaglio il programma sviluppato in Matlab per creare una tipologia specifica di esercizio definita Variable Amplitude.

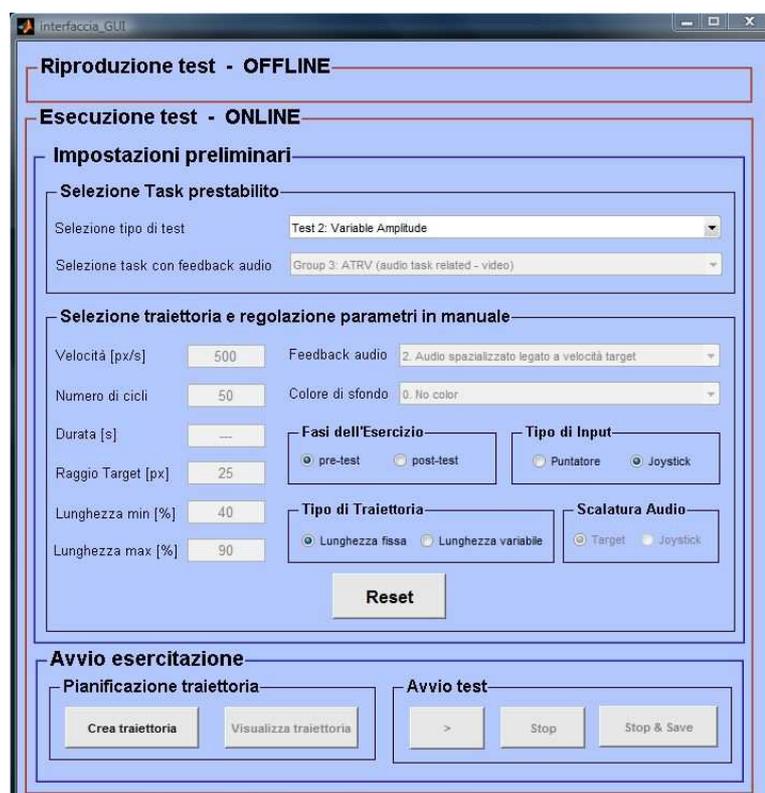


Figura 2.7: Interfaccia GUI realizzata

Questa interfaccia fornisce un appropriato ambiente di lavoro per l'utente che deve gestire le varie fasi della routine dei test (creazione, visualizzazione, avvio e memorizzazione dell'esercizio) e per il soggetto che svolge l'esercizio; inoltre gestisce la comunicazione con il software di generazione del feedback audio (PureData) fornito durante gli esercizi di inseguimento. In particolare, l'interfaccia grafica deve:

- consentire la creazione e la visualizzazione, da parte dell'operatore, di un esercizio riabilitativo, con l'eventuale scelta di tutti i parametri che caratterizzano il task (tipo di traiettoria, dimensioni della traiettoria, dimensioni del target, velocità del target)

- fornire un'ambiente di 'realtà virtuale', dotato di feedback visivo e uditivo, al soggetto che sta eseguendo l'esercizio
- consentire all'operatore di avviare e interrompere l'esecuzione dell'esercizio
- consentire la memorizzazione dei dati relativi ad un esercizio
- gestire l'invio di pacchetti ad un'applicazione esterna deputata alla generazione del feedback audio.

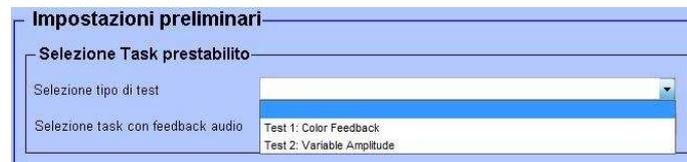
La GUI sviluppata permette un facile approccio tra la macchina e l'operatore e si è scelto di guidare quest'ultimo passo passo in ciascuna fase di selezione del task motorio.

La schermata iniziale prevede la selezione del test a cui si vuol sottoporre il soggetto tramite un menù a tendina(Fig.2.8(a)). Selezionando il test 2: *Variable Amplitude*, ciascun soggetto reclutato sarà sottoposto al medesimo esercizio di target tracking e da un ulteriore menù a tendina sarà possibile scegliere quale feedback audio applicare al tester in esame (Fig.2.8(b)). Per gli obiettivi d'analisi pensati a priori, la selezione delle quattro diverse tipologie di feedback crea fin dall'inizio una suddivisione dei soggetti nei quattro gruppi precedentemente accennati.

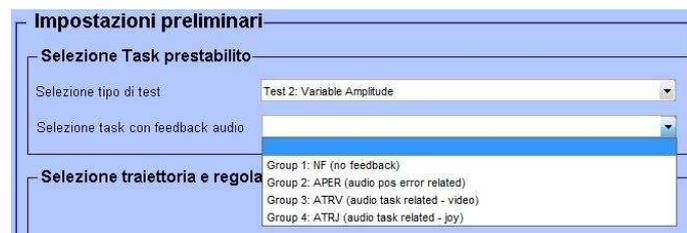
Una volta impostato il test e il tipo di feedback audio applicato, che identifica l'appartenenza ad un gruppo ben definito, sull'interfaccia vengono caricate automaticamente le impostazioni preliminari che caratterizzano la fase on-line del test. Tali impostazioni sono state fissate a priori, dopo aver ripetutamente testato 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 (2.9), numerosi sono i parametri che la caratterizzano.

Nel dettaglio, tali parametri 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:
 - per il Pre_Test sono 50 cicli



(a) Selezione test da sottoporre al soggetto



(b) Identificazione del gruppo a cui associare il soggetto

Figura 2.8: Prime fasi di caricamento del test mediante l'interfaccia GUI

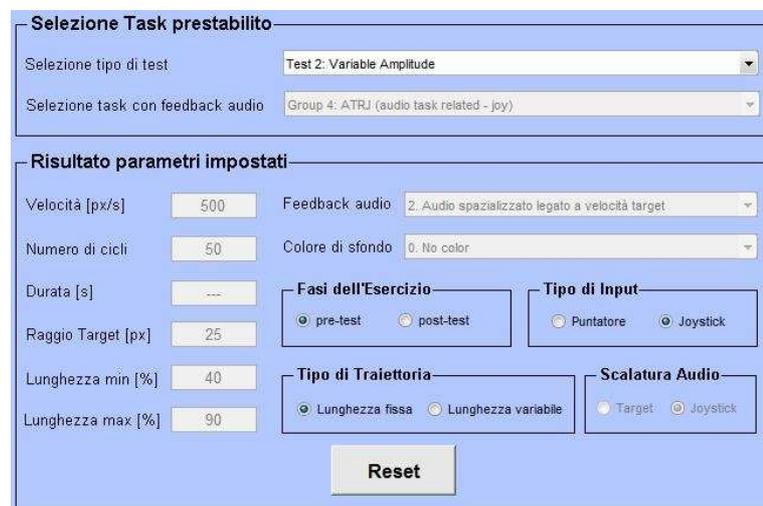


Figura 2.9: Parametri preimpostati una volta selezionato il gruppo di appartenenza

- per il Post_Test sono 40 cicli
- per il Variabile sono 30 cicli
- Durata del task: questo parametro viene semplicemente visualizzato per informare l'operatore sulla precisa durata in secondi dell'esercizio e non

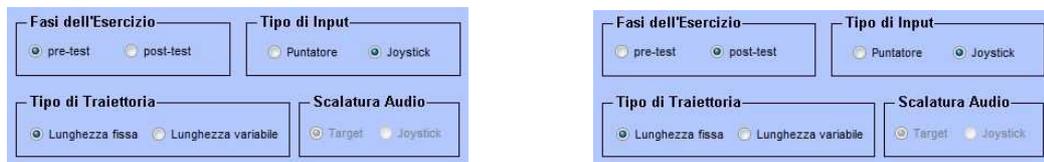
può essere impostato e/o modificato. Il tempo di esecuzione si riferisce ad una larghezza di movimento in x di 1920 pixel, con una velocità impostata a 500 pixel/secondo. Nel nostro Test, abbiamo durate diverse per le tre fasi:

- per il Pre_Test è di 165.80 secondi
 - per il Post_Test è di 132.10 secondi
 - per il Variabile è di 79.96 secondi
- 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, fissata al 90% della lunghezza massima dello schermo per ciascuna fase del test
 - Lunghezza minima che varia in base alla fase del test considerato:
 - per il Pre_Test e il Post_Test non è prevista
 - per il Variabile è pari al 40%: significa che il target termina ogni singolo ciclo in un punto interno all'intervallo compreso fra lunghezza minima e lunghezza massima
 - 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 nel nostro test non è previsto e rimane nero di default

Una volta considerati i valori dei parametri prefissati, l'utente può sottoporre il soggetto ad una delle 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 2.10(a))
- Fase di Post_Test (Figura 2.10(b))
- Fase Variabile (Figura 2.10(c))



(a) Impostazioni per fase di Pre_Test

(b) Impostazioni per fase di Post_Test



(c) Impostazioni per fase Variabile

Figura 2.10: Impostazioni delle diverse fasi del test *Variable Amplitude*

Oltre ai parametri considerati finora, l'interfaccia GUI sviluppata permette inoltre all'utente di selezionare il dispositivo d'ingresso con cui far eseguire l'esercizio al soggetto e a quale aspetto applicare la modalità di feedback audio impostata precedentemente (2.11). In entrambi i casi la selezione avviene mediante radio-botton, dove:

- Nel primo caso, il possibile dispositivo di ingresso con il quale far lavorare i soggetti può essere uno fra:
 1. un joystick a due gradi di libertà (2.2)
 2. un puntatore, rappresentato da un mouse o da una tavoletta grafica (utilizzata nei test precedenti)



Figura 2.11: Selezione del dispositivo di input

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



Figura 2.12: 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 “Reset” è possibile reimpostare i parametri in questione.

Ad ogni modo, una volta caricati i parametri è possibile creare [2.13(a)] e visualizzare [2.13(b)] la traiettoria che il target di riferimento effettuerà a video (Figura 2.13).



(a) Creazione traiettoria



(b) Visualizzazione traiettoria

Figura 2.13: Pianificazione Traiettoria

I parametri sono stati impostati, la traiettoria è stata creata e anche visualizzata; arrivati a questo punto è possibile avviare il test mediante la finestra di avvio (Figura 2.14).

Al termine dell’esecuzione è possibile salvare i dati relativi all’esercizio svolto mediante il tasto “Stop & Save”, che permette un salvataggio nella cartella di riferimento.



Figura 2.14: Finestra di avvio e di salvataggio

Le impostazioni di salvataggio sono di default:

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

alle quali, volta per volta, basta inserire “*Cognome*” e “*Nome*” del soggetto testato.

2.4 Implementazione dell’interfaccia GUI

L’interfaccia GUI gestisce le varie fasi della creazione ed esecuzione del task permettendo la comunicazione tra Matlab, Simulink e il PureData. In particolare, essa coordina le funzioni che creano ed impostano la traiettoria con la funzione *draw*, la quale aggiorna le variabili relative al soggetto ed al target, e la funzione *eventListener* che permette di eseguire le istruzioni contenute nella *draw* in maniera ripetitiva e sincronizzata.

Creazione della traiettoria

Punto di partenza è la funzione *build_trajectory2*, che crea gli estremi della traiettoria: questa è rappresentata da un vettore contenente in ogni cella la coordinata x in cui avviene il cambio di direzione. Queste informazioni sono sufficienti per definire la traiettoria in quanto lo spostamento lungo l’asse y è sempre nullo.

La posizione di partenza è la coordinata centrale lungo l’asse x , ogni movimento ripassa per questa posizione e continua per una lunghezza random, senza eccedere i limiti impostati. Tale funzione necessita in entrata dei seguenti parametri:

- n : dimensione del vettore traiettoria
- min : lunghezza minima del ciclo rispetto a *starting_position*
- max : lunghezza massima del ciclo rispetto a *starting_position*
- *starting_position*: posizione di partenza del target lungo l’asse x

La funzione *traj_pp* pianifica poi la traiettoria punto per punto, interpolando i valori assegnati da *build_trajectory2* per le variabili di posizione iniziale e finale di ogni ciclo dell'esercizio, rispettando i vincoli imposti sui punti estremi e sul tempo di transizione. Come funzione interpolatrice è stata utilizzata una funzione polinomiale di quinto grado

$$q(t) = a_0 + a_1t + a_2t^2 + a_3t^3 + a_4t^4 + a_5t^5 \quad (2.3)$$

che specifica istanti di tempo iniziali e finali, oltre a posizioni, velocità ed accelerazioni. 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 di regolarità della traiettoria che corrisponde al criterio di 'minimum-jerk' (minimo valore quadratico medio della derivata dell'accelerazione).

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 di tale struttura 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* richiama la funzione *draw* durante l'esercizio.

Altri parametri che vengono salvati in '*exercise_parameters*', oltre al '*.data*' 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 (descrizione dell'esercizio)
- grado della funzione interpolatrice
- etichette che identificano le variabili di *exercise_parameters.data*
- tipo di test
- tipo di task
- istanti di inizio ciclo.

2.4.1 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 3.2.1).

La scalatura nella fase Variabile non avviene in un ciclo esatto ma si modifica in modo continuo ad ogni ciclo. Per questo motivo 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

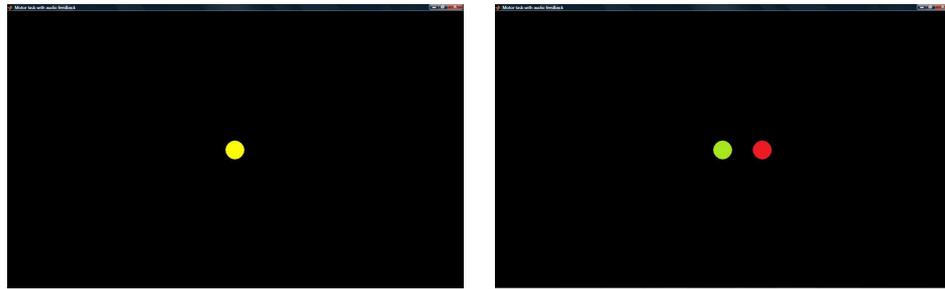
in questo modo ad ogni cambio di direzione il fattore di scalatura permette di agire di conseguenza.

2.4.2 Aggiornamento delle variabili: Funzione Draw

Soffermandosi alla raffigurazione dei target realizzata 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 2.15(a)) ma i continui aggiornamenti sulle coordinate di posizione dei target avvengono, scanditi dall’*eventListener*, nella funzione *draw*(Figura 2.15(b)).

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



(a) Inizializzazione dall'interfaccia GUI

(b) Aggiornamenti grazie alla draw

Figura 2.15: WorkSpace

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 2.16) 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

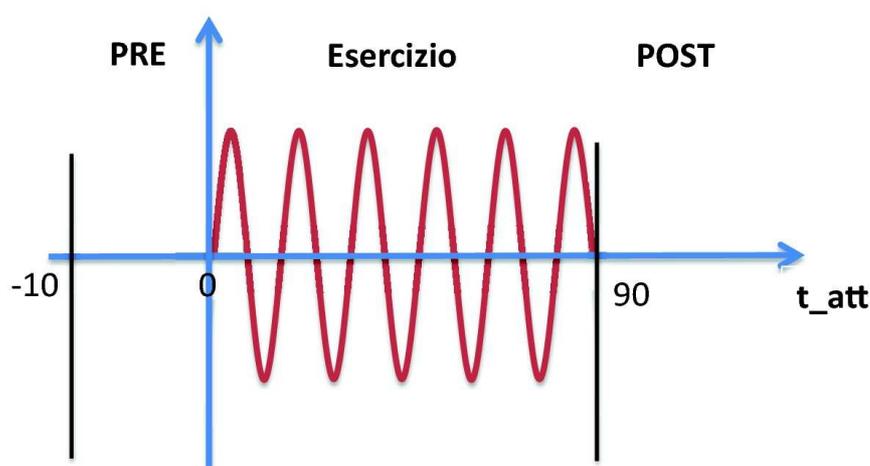


Figura 2.16: Relazioni tra i vari tempi utilizzati all'interno della draw

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 (end).

Il primo passo consiste nel disegnare l'indicatore del target (pallino rosso) e del soggetto (pallino verde); quest'ultimo è realizzato sfruttando le coordinate di posizione presenti nella struttura `sim_data` presente in ingresso. Come già detto, l'interfaccia prevede la possibilità di applicare un'eventuale scalatura video alle

coordinate delle posizioni ed è proprio la *draw* che se ne occupa, moltiplicando tali coordinate per il fattore che ne riduce o amplifica la scala.

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, ovviamente prima che il target da seguire inizi a muoversi. L'avvio dell'esercizio è preceduto da 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 campi a zero tranne quello relativo all'audio, il quale è composto da una successione correttamente temporizzata di 0 e 9.

Durante la fase di esercizio, i target vengono disegnati a schermo grazie ad apposite istruzioni contenute nella funzione in questione (*draw*) e ciò avviene ogni qualvolta essa viene rieseguita. In questo modo si ha un continuo aggiornamento a video dei target, in modo da poter visualizzare al meglio il loro movimento (che appare continuo). Le istruzioni che permettono questa funzionalità sono:

```
set(hg_obj_tar, 'xData', xtarV + offset(3)/2, 'yData', ytarV + offset(4)/2);
```

per il target rosso, cioè quello che il soggetto deve seguire. Mentre per il target verde, che corrisponde al paziente abbiamo:

```
set(hg_obj_pat, 'xData', xy_patV(1) + offset(3)/2, 'yData', xy_patV(2) + offset(4)/2);
```

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 (non nel nostro caso), 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 contenente 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:

```
vstar\_xV = vstar\_x(indt)*fatt\_t(indt)
vstar\_yV = vstar\_y(indt)*fatt\_t(indt)
```

e sulla posizione del target del soggetto:

```
xstarV = xstar(indt)*fatt\_t(indt)
ystarV = ystar(indt)*fatt\_t(indt)
```

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 e 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 può 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 può passare all’analisi dei pacchetti che l’interfaccia invia al software esterno (Pure Data) per la generazione del tipo di audio desiderato. Il dato da inviare, come descritto in precedenza, 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)]);
```

L’etichetta `audio_sel` specifica il tipo di audio in uso durante l’esercizio, che può essere uno di quelli riassunti nella seguente tabella (tab. 2.1).

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:

```
[~, datastring] = cstruct(pd_packet)
```

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. I dati vengono convertiti in `single` prima

di essere passati a cstruct in quanto in questo modo viene rispettato IOSC Arguments che richiede un numero a virgola mobile a 32 bit e l'allineamento a 4 byte. 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. 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:

Tabella 2.1: Flag utilizzati per identificare i diversi feedback audio.

flag	tipo di feedback audio
0	no audio
1	velocità del target non spazializzato
2	velocità del target spazializzato
3	errore di posizione non spazializzato
4	errore di posizione spazializzato
5	errore di velocità non spazializzato
6	errore di velocità non spazializzato
7	non definito
8	beep di fine esercizio
9	beep di inizio esercizio

```
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, sotto forma 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.

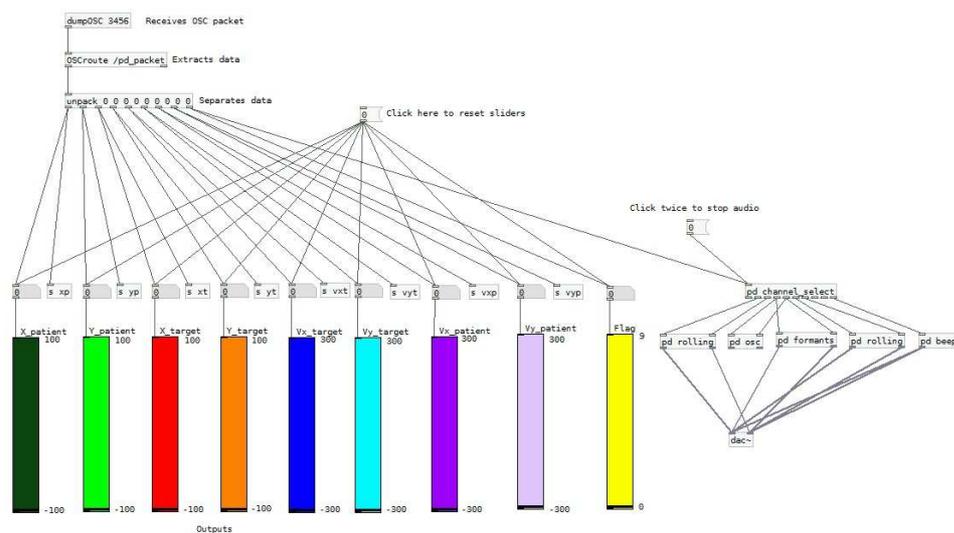


Figura 2.17: PureData: le slider sulla sinistra rappresentano i valori ricevuti in ingresso, mentre i blocchetti sulla destra identificano i diversi tipi di feedback (pd rolling (1) = ATR, pd formats = APER, pd rolling (2) = AVER, pd beep = beep di inizio e fine esercizio)

2.5.5 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.

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.

2.6 Feedback visivo

La riproduzione dell'esercizio avviene in una finestra dedicata; questa viene aperta, a schermo intero, su un monitor Full HD, risoluzione 1080x1920 sul quale

sono visibili le posizioni di target e soggetto e attraverso il quale viene fornito il feedback visivo. Nello specifico, gli indicatori sono 2 oggetti grafici di tipi line, creati all'interno della funzione creaFigura, attraverso le seguenti istruzioni:

```
hg_obj_tar=line('Parent',ax,'XData',[],'YData',[],'Color','k','EraseMode','xor','Marker','o','MarkerSize',40,'MarkerFaceColor','r');

hg_obj_pat=line('Parent',ax,'XData',[],'YData',[],'Color','k','EraseMode','xor','Marker','o','MarkerSize',40,'MarkerFaceColor','g');
```

La proprietà 'Parent' definisce gli assi (ax) a cui tali oggetti fanno riferimento, mentre 'XData' e 'YData' identificano i vettori delle coordinate degli indicatori lungo l'asse x e l'asse y , la cui terna di riferimento è posta al centro dello schermo. Tale traslazione è effettuata prendendo le coordinate della finestra rispetto alla terna di riferimento dello schermo e aggiungendo un offset lungo x e lungo y pari a metà delle dimensioni x e y di tale finestra.

- `get(0,'MonitorPositions')` indica la dimensione dello schermo, la terna di riferimento è posizionata in basso a sinistra
- `get(1,'Position')` restituisce le coordinate della finestra dell'esercizio rispetto all'origine dello schermo
- `xy = [xy_pat(1)-offset(3)/2 xy_pat(2)-offset(4)/2]` toglie l'offset centrando la terna di riferimento.

I campi 'Marker', 'MarkerSize', 'MarkerFaceColor' impostano rispettivamente forma, dimensione e colore dell'oggetto; 'Color' definisce il colore della retta su cui si muove l'oggetto, 'EraseMode' fa in modo che il colore dell'oggetto non si sovrapponga al colore dello schermo.

Capitolo 3

Test Sperimentali

Il presente studio rientra in un progetto più ampio che il gruppo di robotica del DIMEG dell'Università di Padova porta avanti da alcuni anni [7] [8] [9] [10]: si è cercato di approfondire il ruolo che il feedback audio gioca nel motor learning e 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.

3.1 Test precedenti

Il punto di partenza di questo lavoro è un progetto precedente [[48] e [49]], dove 3 tipi di feedback audio sono stati sviluppati e forniti online durante l'esecuzione di 6 esercizi di target tracking: (1) assenza di audio, (2) feedback audio legato alla velocità del target (task-related) e (3) feedback audio legato all'errore di posizione del soggetto rispetto al target (position error-related); ciascun tipo di feedback è stato testato in una prova a traiettoria a lunghezza fissa e in una dove la lunghezza della traiettoria percorsa dal target variava in maniera random. Uno degli obiettivi è stato quello di verificare se la somministrazione continua del feedback audio task-related potesse essere più efficace di un feedback error-related in termini di

prestazioni, durante l'esecuzione di un task di tracking complesso. Si è ritenuto che il feedback audio task-related potesse fornire informazioni utili al soggetto per migliorare la performance più di quanto potesse fare il feedback position error-related. Come device era stata utilizzata una tavoletta grafica (figura 3.1).

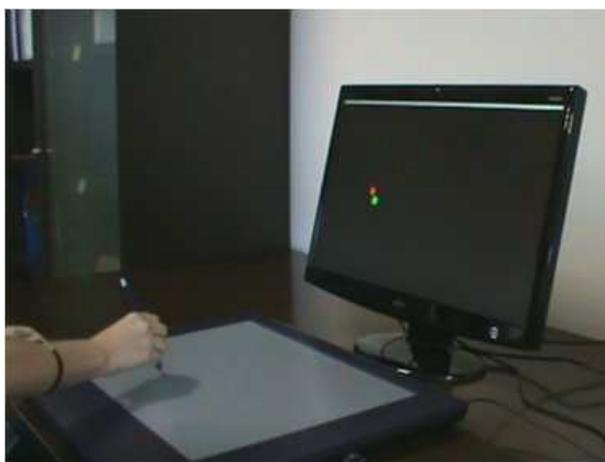


Figura 3.1: Tavoletta grafica utilizzata nei test precedenti

Questi sono i principali risultati raggiunti [48]:

- La figura 3.2 (sopra) mostra la media pesata dell'errore di tracking nei vari task, normalizzato rispetto al raggio del target. L'analisi statistica evidenzia che, considerando la stessa modalità di feedback audio, non ci sono differenze rilevanti tra il task a lunghezza fissa e variabile. Comunque, il task C e D presentano entrambi un errore significativamente più basso rispetto ai task A e B, mentre nei task E ed F la presenza di un feedback error-related non migliora in modo sostanziale la performance rispetto al caso in cui non vi è feedback (task A).
- L'analisi statistica sull'integrale della velocità relativa R_{vel} (figura 3.2 (sotto)) mostra, come previsto, che i task a lunghezza fissa sono sempre eseguiti meglio dei corrispondenti a lunghezza variabile, indipendentemente dalla modalità di feedback considerata. Tuttavia, non sembrano essere emersi significativi miglioramenti quando il feedback audio è fornito nella stessa modalità di esercizio (lunghezza fissa o variabile).

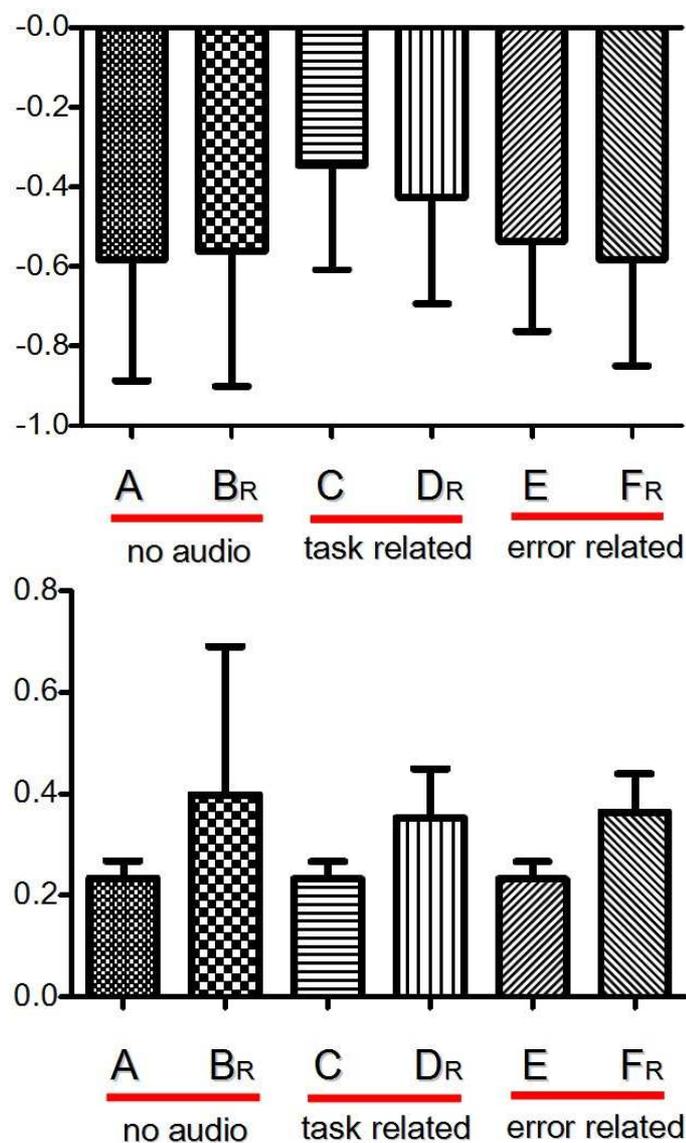


Figura 3.2: sopra: media pesata dell'errore di tracking (normalizzato rispetto al raggio del target); sotto: integrale della velocità relativa (normalizzato rispetto alla lunghezza del tratto). Task A (lunghezza fissa - no audio), B (lunghezza casuale - no audio), C (lunghezza fissa, audio task related), D (lunghezza casuale - audio task related), E (lunghezza fissa - error related), F (lunghezza casuale - error related).

3.2 Definizione del test

Sulla base di tali risultati, questa tesi si propone di operare un confronto tra diversi feedback uditivi, principalmente ottenuti da quelli appena descritti, ma applicati a diverse situazioni visive. Essi forniranno informazioni sulle stesse grandezze (errore di velocità e velocità del target), con lo scopo di capire in quale modo essi possano dare un contributo per una maggiore accuratezza nell'inseguimento del target.

3.2.1 Fasi del Test

Per la sessione di test in questione sono state suddivise le persone a disposizione (vedi cap.3.3.2) in quattro gruppi distinti. Qualunque soggetto che ha partecipato all'esercizio, indipendentemente dal gruppo di appartenenza, è stato sottoposto a tre fasi distinte del test definite per praticità:

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

L'ordine di esecuzione di tutto il test è scandito seguendo *obbligatoriamente* la successione delle fasi, proprio come sono appena state elencate.

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:

- i) NF-NoFeedback
- ii) APER-AudioPositionErrorRelated
- iii) ATRV-AudioTaskRelatedVideo
- iv) ATRJ-AudioTaskRelatedJoystick

Ogni fase del test è costituita da una serie di movimenti di 'target tracking' orizzontali in direzione alternata destra/sinistra e l'esercizio inizia e termina nella stessa posizione al centro dello schermo (vedi esempio di tracciato in fig. 3.3).

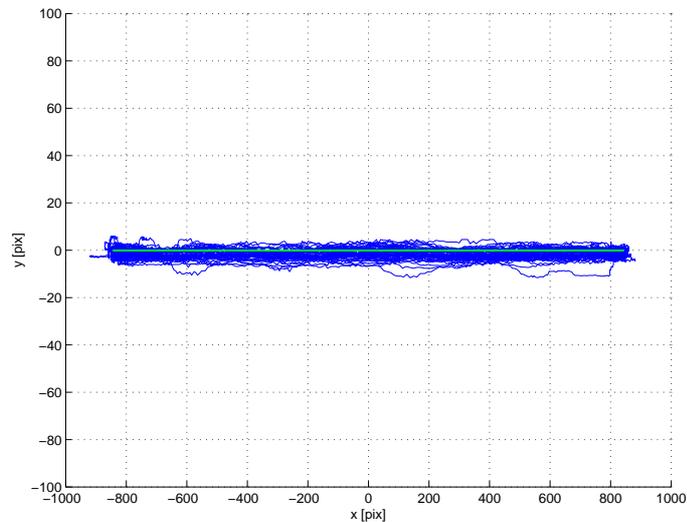


Figura 3.3: Grafico relativo al percorso del target (in verde) e del soggetto (in blu) lungo l'intero task, sull'asse x e y .

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 una minima conoscenza dello strumento di lavoro.

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 3.4(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 3.4(b)), ovvero fino alla fine della fase di Pre_test.

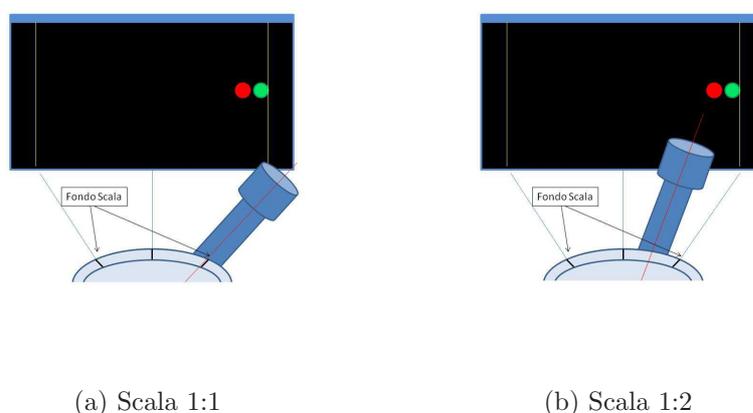


Figura 3.4: Variazione scalatura al 25° ciclo nella fase di Pre_Test

In tale fase (Pre_test) si verifica un particolare comportamento del soggetto: esso 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.

La modifica nel movimento effettuato dal soggetto con il joystick è visibile in figura 3.5:

- nel grafico in alto si vede in modo chiaro come i primi 24 cicli (che corrispondono ad un intervallo temporale di 80 secondi circa) abbiano la scala del joystick totalmente sovrapponibile a quella costante del video, cioè il movimento effettuato dal soggetto corrisponde a ciò che viene visualizzato su schermo. Invece a partire dal 25° ciclo fino al termine del task, la scala del soggetto raddoppia e quindi il movimento che il soggetto compie per effettuare lo stesso movimento è in realtà minore rispetto alla prima parte dell'esercizio; ciò si vede ampiamente analizzando la differenze tra scala del soggetto (blu) e scala del video (verde).
- nel grafico in basso invece si evidenzia la visualizzazione della scala reale, per cui il movimento del soggetto è sempre sovrapposto al movimento visibile

su schermo, anche nella parte finale dell'esercizio, dove in realtà avviene il cambio della scala.

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 di rilassarsi ed eventualmente di togliersi le cuffie, nel caso gli dessero fastidio. Si è scelto una tale durata per la pausa (5 minuti) poichè corrisponde ad un tempo che si trova spesso in letteratura. Durante tale pausa venivano chiusi sia Matlab che Pure Data (vedi cap.3.4.1).

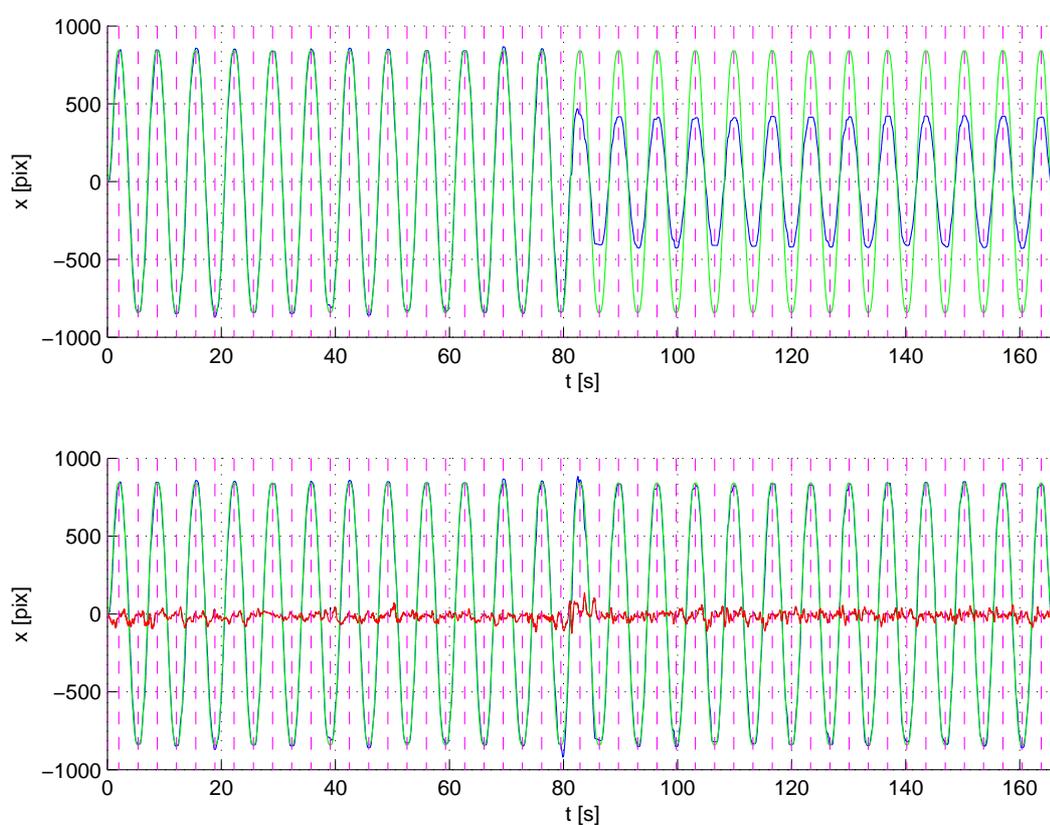


Figura 3.5: Confronto tra la traiettoria su scala reale del joystick (sopra) e quella su scala video (sotto). Il tracciato verde è legato al percorso del target da seguire, infatti in entrambi i grafici è costante e pari al valore massimo. Il tracciato blu indica invece il percorso effettuato dal soggetto, che nel primo grafico cambia dalla metà in poi (a causa del cambio scala sul joystick), mentre nel secondo è sempre costante, poichè su schermo vedo sempre il movimento del target del soggetto che ha ampiezza costante.

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 realizzata nella parte finale della fase di Pre_Test, ovvero scala 1:2 (Figura 3.6(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 3.6(b)), ovvero fino alla fine della fase di Post_test. Anche in questa

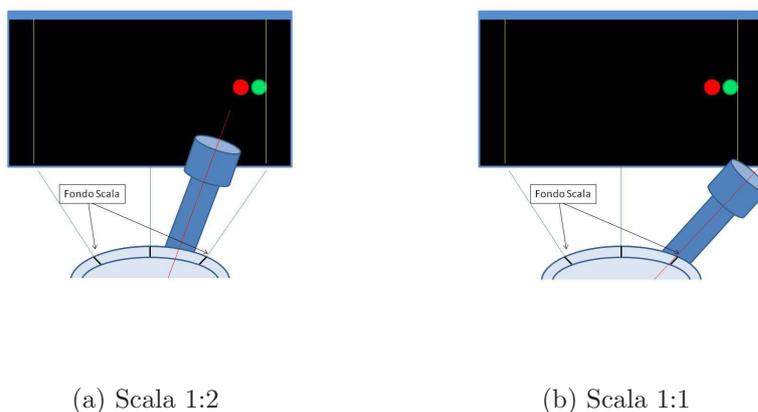


Figura 3.6: Variazione scalatura al °21 ciclo nella fase di Post_Test

fase il soggetto deve modificare il proprio comportamento: per seguire il target 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.

La modifica nel movimento effettuato dal soggetto con il joystick è visibile in figura 3.7:

- nel grafico in alto si vede in modo chiaro come i primi 20 cicli (che corrispondono ad un intervallo temporale di 70 secondi circa) abbiano la scala del soggetto raddoppiata e quindi il movimento che il soggetto compie per effet-

tuare il solito spostamento è in realtà minore rispetto a quello massimo; ciò si vede in modo chiaro analizzando la differenza tra scala del soggetto (blu) e scala del video (verde). Invece a partire dal 21° ciclo fino al termine del task, la scala del joystick torna ad essere totalmente sovrapponibile a quella costante del video, cioè il movimento effettuato dal soggetto corrisponde a ciò che viene visualizzato su schermo.

- nel grafico in basso invece si evidenzia la visualizzazione della scala reale, per cui il movimento del soggetto è sempre sovrapposto al movimento visibile

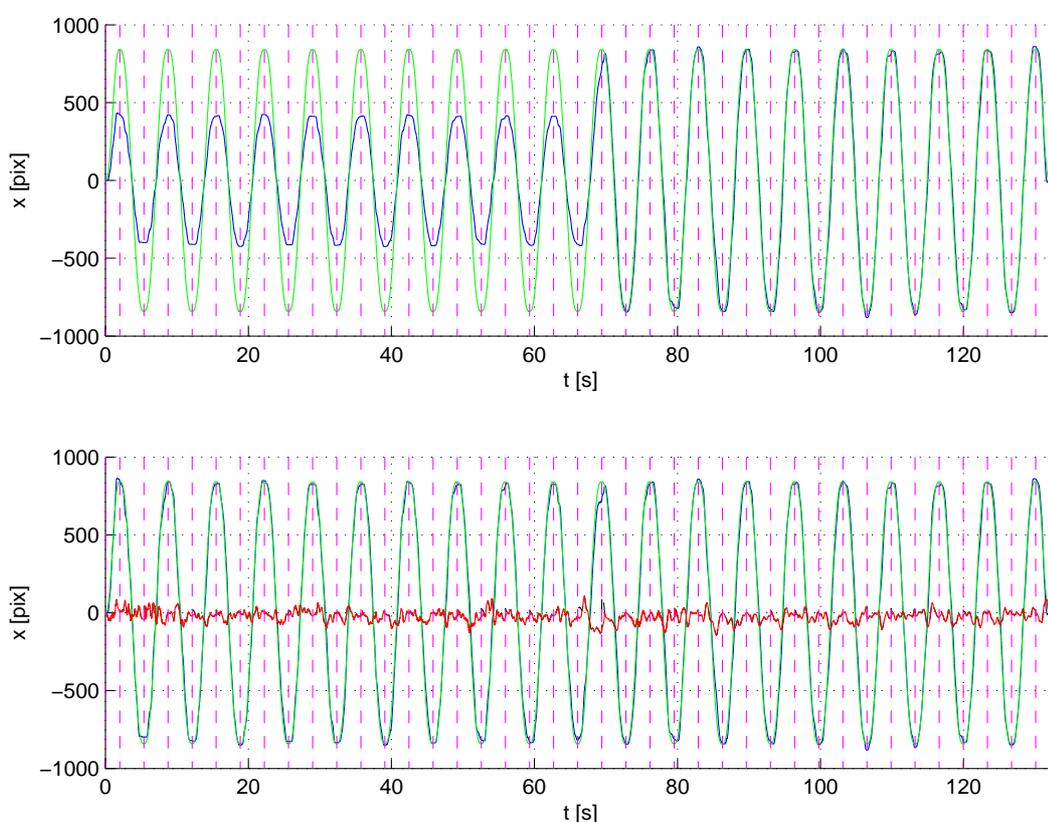


Figura 3.7: Confronto tra la traiettoria su scala reale del joystick (sopra) e quella su scala video (sotto). Il tracciato verde è legato al percorso del target da seguire, infatti in entrambi i grafici è costante e pari al valore massimo. Il tracciato blu indica invece il percorso effettuato dal soggetto, che nel primo grafico è minore rispetto all'andamento normale (a causa del cambio scala sul joystick), mentre nel secondo è sempre costante, poichè su schermo vedo sempre il movimento del target del soggetto che ha ampiezza costante.

su schermo, anche nella parte iniziale dell'esercizio, dove in realtà esiste una scala variata.

Tra la fase di Post_Test e quella Variabile l'unica pausa concessa al soggetto è stata quella necessaria per la chiusura di Matlab e Pure Data (vedi cap.3.4.1), poi il soggetto veniva sottoposto subito alla terza e ultima fase del test.

Fase Variabile

Nell'ultima fase del test sono stati previsti una totalità di 30 cicli. A differenza dei due test precedenti (Pre_Test e Post_Test), l'ampiezza dei cicli di questa fase 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*.

Come si può vedere nell'immagine (Fig.3.9), nella parte superiore, il grafico rappresenta l'effettiva traiettoria visualizzata a video del target di riferimento nella fase di traiettoria a lunghezza variabile.

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.

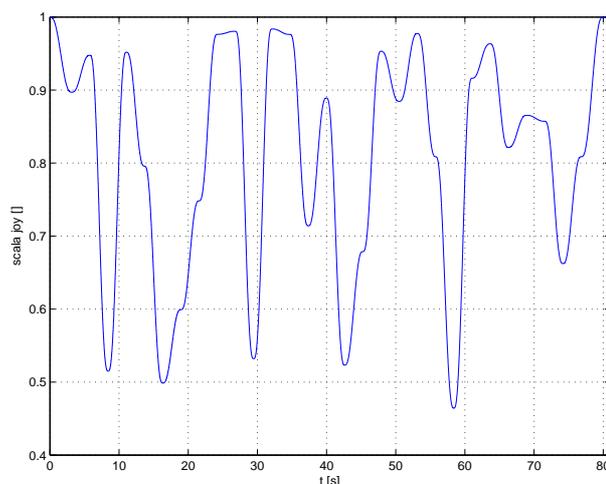


Figura 3.8: Progressiva scalatura del joystick, in funzione dell'ampiezza di ogni ciclo, avendo una traiettoria a larghezza variabile.

Il fattore di scalatura è stato calcolato nei punti estremi di ciascun movimento come rapporto tra l'ampiezza del movimento del target e l'ampiezza massima. In ciascun tratto del movimento è stato usato un fattore di scalatura, variabile tra i due valori estremi e calcolato utilizzando un polinomio interpolatore di quinto grado (vedi eq. 2.3).

Nella figura 3.8 viene visualizzata la progressiva scalatura del joystick, in funzione di ogni lunghezza del ciclo impostata dalla traiettoria variabile.

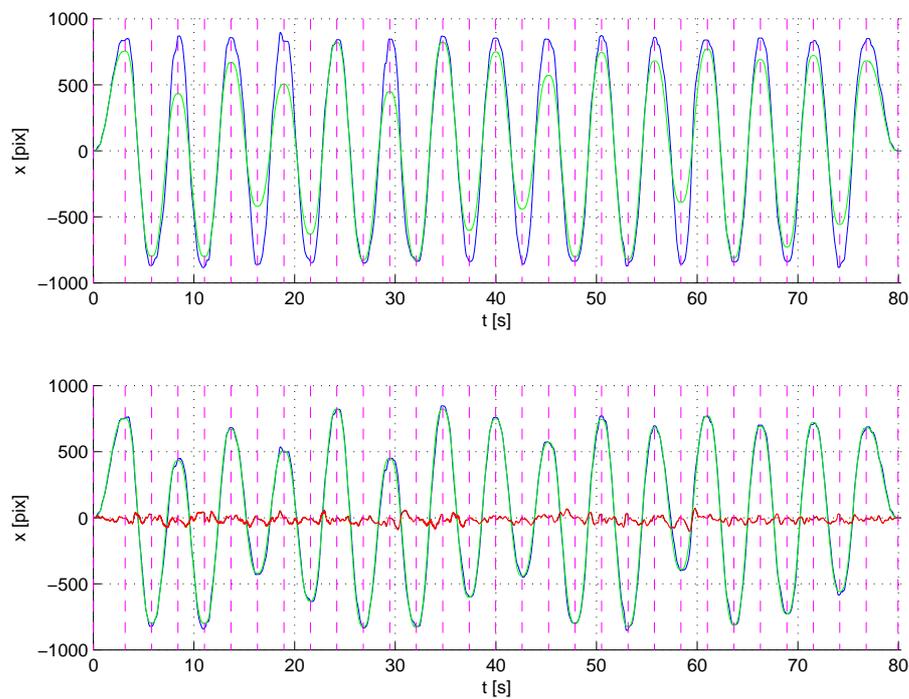


Figura 3.9: Confronto tra la traiettoria su scala reale del joystick (sopra) e quella su scala video (sotto). Il tracciato verde è legato al percorso del target da seguire e in entrambi i grafici ha un andamento a larghezza variabile. Il tracciato blu indica invece il percorso effettuato dal soggetto: nel primo grafico è sempre costante, infatti con il joystick si effettua sempre uno stesso movimento a larghezza massima; nel secondo invece segue l'andamento del tracciato verde, infatti a video il movimento del soggetto sembra variare di volta in volta.

3.2.2 Tipologie Feedback Audio

Sebbene molte evidenze sperimentali supportino 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 progettato in modo errato 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.

Per la sintesi del seconda modalità audio (errore di posizione) è stata usata come base la versione precedentemente sviluppata, alla quale sono state apportate alcune modifiche per ottenere un risultato migliore. La modalità di partenza realizzava una sintesi vocale che spaziava su tutte le vocali dell'alfabeto, gli ingressi

non erano normalizzati e il suono era legato all'errore di posizione su entrambe le dimensioni x e y del piano; la versione utilizzata in questo caso, invece, sfrutta ingressi normalizzati considerando solo l'errore di posizione nelle coordinate in x , mappato sulle frequenze del suono, la sintesi vocale si limita all'uso di 2 vocali (A e U), dove il passaggio dalla prima alla seconda avviene per errori molto grandi; inoltre, è stata prevista una soglia di saturazione. L'errore di posizione in x è definito come:

$$\epsilon = |x_s - x_t| \quad (3.1)$$

Spazializzazione del suono

Entrambe le tipologie di audio sono disponibili in due modalità: spazializzata e non spazializzata. Per spazializzazione si intende un insieme di tecniche per l'elaborazione dei segnali sonori attraverso le quali una sorgente virtuale di suono può essere posizionata in un qualunque punto dello spazio attorno all'ascoltatore. La spazializzazione delle informazioni audio incrementa il realismo dell'ambiente virtuale di lavoro, portando alla creazione di un vero e proprio un paesaggio sonoro (soundscape) tridimensionale all'interno del quale le sorgenti assumeranno una posizione diversa rispetto all'ascoltatore e potranno anche 'muoversi' intorno ad esso più o meno velocemente. La spazializzazione delle tipologie di audio rende il feedback sonoro, fornito durante l'esecuzione degli esercizi, dipendente anche dalla posizione del target, valutata nella coordinata x dello schermo.

Suddivisione gruppi audio

È già stato specificato in precedenza come l'appartenenza a gruppi distinti è funzione del tipo di feedback audio a cui i soggetti sono stati sottoposti. Perciò definendo i diversi tipi di audio che si intendono utilizzare, si vengono a creare in modo automatico le diverse suddivisioni dei soggetti.

Per quanto riguarda questo test, sono state applicate 4 situazioni sonore differenti e quindi abbiamo i 4 gruppi:

- i) **NF-NoFeedback**: al soggetto non viene fornito alcun effetto sonoro durante tutto il task, quindi l'esecuzione si baserà solamente sull'utilizzo del feedback visivo dei pallini in movimento.
- ii) **APER-AudioPositionErrorRelated**: in questo caso il soggetto utilizza un feedback audio relativo all'errore di posizione (come definito precedentemente), cioè un effetto sonoro che indica l'errore di posizione del suo target (pallino verde) rispetto al target che deve inseguire (pallino rosso).
- iii) **ATRV-AudioTaskRelatedVideo**: il feedback audio utilizzato in questo caso è quello relativo alla velocità del target da inseguire, cioè del pallino rosso. La velocità presa in considerazione è quella relativa al movimento del pallino visualizzata a video.
- iv) **ATRJ-AudioTaskRelatedJoystick**: anche in questo gruppo l'audio utilizzato è il tipo *task related*, cioè l'effetto sonoro è legato alla velocità del target. La velocità presa in considerazione è sempre quella del movimento effettuato dal target rosso (da seguire), però, a differenza del gruppo ATRV, essa viene riportata alla scala del joystick.

3.3 Popolazione

Per la tipologia di test considerato si è scelto di reperire un numero di soggetti abbastanza sostanzioso, 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 (vedi Sez. 3.3.1), a cui rispondere in modo chiaro e veritiero. 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 ha 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 sono conclusi il 22 Gennaio 2011; sono 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. Inoltre è stata regolata l'illuminazione sul display per evitare i riflessi della luce del sole, sono stati staccati tutti i telefoni presenti all'interno della stanza e durante l'esecuzione dei test non era permesso a nessuno di entrare.

Nelle diverse fasi dell'esercizio i soggetti sono stati minuziosamente monitorati per cercare di rilevare se grossolani errori nel target tracking fossero condizionati da particolari agenti esterni, magari non considerati a priori (come ad esempio un rumore esterno, un cambio di impugnatura, ecc.).

3.3.1 Questionario

Ai soggetti che si sono offerti di sostenere il test è stato chiesto di compilare un questionario, in modo che le loro risposte potessero essere un'utile integrazione da prendere in considerazione durante l'analisi dei dati.

Le domande presentate nel questionario sottoposto riguardano: i dati anagrafici del soggetto (nome, cognome, data di nascita, luogo di nascita, luogo di residenza), alcuni dati legati a possibili problemi fisici (problemi di vista specificando anche l'entità, l'utilizzo di lenti o occhiali, problemi di udito, problemi all'arto superiore usato per il test) e 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).

Dalle risposte ottenute, siamo stati in grado di reperire alcune informazioni utili a verificare successivamente l'influenza di alcuni fattori personali nell'esecuzione dei test, ai fini dell'analisi dei dati.

I questionari non hanno evidenziato alcun caso particolare e le persone contattate sono state tutte selezionate per il test: è emerso che la maggior parte di esse praticano sport (36/47), quasi nessuno suona uno strumento musicale (8/47), i soggetti in media passano 2,46h al giorno davanti al pc e 4,1h al giorno davanti ad un qualsiasi schermo. Inoltre, sapendo quali persone possiedono difetti alla vista, in fase di test di è prestata attenzione che quest'ultimi utilizzassero gli occhiali o le lenti per non avere una prova falsata.

3.3.2 Gruppi ottenuti mediante randomizzazione

I soggetti reperiti per il test sono stati suddivisi mediante un programma di randomizzazione, creato appositamente per realizzare al meglio i 4 gruppi. Con l'ausilio di questo programma, si è cercato di distribuire nel miglior modo possibile i maschi e le femmine, come anche i soggetti mancini e i destrorsi, tra i possibili raggruppamenti. I 4 gruppi generati presentano le seguenti caratteristiche:

1. Gruppo1: NF

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

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

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

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 Esecuzione dei test

Per la sessione di test si è cercato di creare una postazione idonea, che potesse ridurre al minimo ogni possibile causa esterna di distrazione. Ogni soggetto è stato fatto accomodare davanti ad una scrivania su cui è stato posto lo schermo; affianco alla sedia (dalla parte del braccio utilizzato), è stato posizionato il joystick, ad un'altezza tale per cui il soggetto aveva il braccio libero da impedimenti e l'angolo del gomito era di circa 90°; sono state inoltre fornite delle cuffie per il feedback audio.

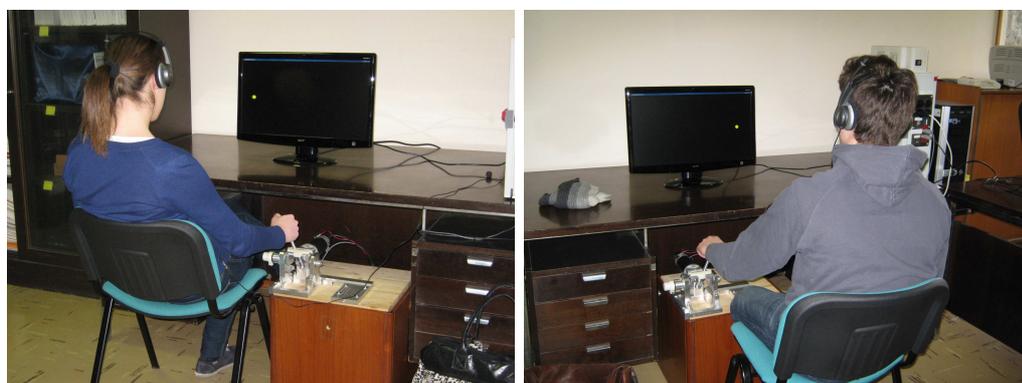


Figura 3.10: Alcuni soggetti durante i test

L'obiettivo richiesto è stato semplicemente quello di seguire, al meglio delle possibilità, il movimento dell'indicatore circolare di colore rosso rappresentato a video. Il soggetto è stato istruito su postura e comportamenti corretti da tenere durante la fase di test; è stato inoltre informato della presenza o meno di un feedback audio, fornito durante l'esecuzione. Prima dell'inizio del test, al soggetto veniva presentato un training di prova, necessario per prendere confidenza con il joystick.

Progressivamente all'esecuzione dei test, è stata compilata una tabella riassuntiva contenente tutte le informazioni riguardanti ogni soggetto, come eventuali situazioni singolari verificatesi nell'esecuzione del test, specificando il task e, se possibile, l'istante di tempo nel quale tale irregolarità si è verificata.

3.4.1 Problemi riscontrati

Ancor prima di sottoporre i soggetti al test sviluppato nel seguente progetto, si è provato ad ipotizzare un possibile insieme di problematiche che nel corso dei test potesse compromettere la corretta valutazione degli esercizi. Nel riflettere su questo, abbiamo notato che l'utilizzo del joystick installato su una cassettera di legno sarebbe potuto essere scomodo da utilizzare da parte di tutti i mancini ed è per questo che si è deciso di porre la cassettera con il joystick alla destra dei destrorsi, mentre ai mancini è stata posizionata di fronte.

Nel caricare il programma elaborato, i problemi principali riscontrati erano legati alla visualizzazione dei target a video, che spesso risultavano tagliati, ma fortunatamente 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, riguarda il fatto che spesse volte Windows era soggetto a crash. Le ipotesi più plausibili per valutare da cosa derivi questo problema 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, mentre 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 l'attività con i tempi pilotati da Simulink stesso e si presume che questo porti ad un crash di Windows.

Dal punto di vista dei problemi legati direttamente ai soggetti sottoposti al test, una delle maggiori evidenze (vista in un buon numero dei partecipanti) era legata alla poca ergonomia della leva del joystick, cosa che per molti ha portato ad 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	O	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	F	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	F	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	M	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 tipologia 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 che il risultato fornito da 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:

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

dove t_1 e t_2 indicano rispettivamente l'istante di inizio e fine ciclo.

La valutazione dell'integrale avviene in maniera approssimata sommando i contributi relativi ad ogni campione pesato con l'intervallo di campionamento (0,01 s). 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 dà un'indicazione sullo spazio percorso inutilmente in ogni ciclo, indipendentemente dall'accuratezza con cui l'esercizio è stato eseguito.

4.1.3 Ritardo medio (emx)

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

$$err_x(t) = (x_s(t) - x_t(t))sign(v_{x_t}(t))(-1)^{c(t)} \quad (4.4)$$

dove la differenza delle posizioni lungo l'asse x fra soggetto e target ($x_s(t) - x_t(t)$) è pesata con la direzione del movimento del target, ovvero con il segno della velocità di quest'ultimo. A questo tipo di errore ci si può riferire anche con il termine ritardo medio poiché si è visto che, mediamente, tale errore in tutti i

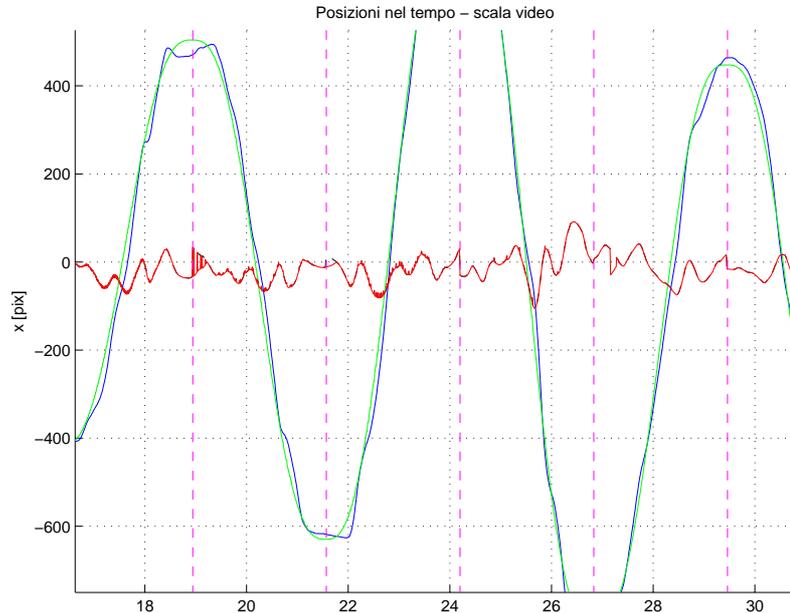


Figura 4.1: Tracciato di un test Variabile, in cui si evidenzia l'andamento dell'errore (linea continua rossa) e la correzione applicata a tale errore (linea tratteggiata nera), in quanto si è rivelato inconsistente nel momento in cui avviene il cambio di direzione da parte del target, cioè quando avviene un cambio nel segno della sua velocità.

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, cioè quando avviene un cambio nel segno della sua velocità (vedi fig. 4.1). In generale il soggetto tende ad essere in ritardo rispetto al target da seguire, pertanto quando quest'ultimo ha già completato il proprio ciclo, il soggetto deve ancora portarlo a termine. Se trascurassimo l'ultimo fattore dell'equazione 4.4, si vede come verrebbe preso in considerazione solo il segno della velocità del target, e nel momento in cui avviene il cambio del ciclo, il soggetto, che sarebbe in ritardo rispetto al target, verrebbe ora considerato in anticipo. Tale situazione si protrae finché i segni delle velocità dei 2 indicatori rimangono discordi. Per ovviare a tale inconveniente è stato quindi introdotto l'ultimo fattore correttivo che si vede nell'equazione 4.4 e applicato nella prima metà di ogni ciclo, dove: con

$$\begin{aligned}
 c(t) &= (v_{x_s}(t) - v_{x_t}(t)) \text{sign}(x_s(t) - x_t(t)) < 0 \\
 &\wedge v_{x_s}(t) \neq v_{x_t}(t) \\
 &\wedge |v_{x_t}(t) - v_{x_t}(t - t_c)| > 0
 \end{aligned} \tag{4.5}$$

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
- $|v_{x_t}(t) - v_{x_t}(t - t_c)| > 0$: stabilisce se il modulo della velocità del target è crescente; in tal modo viene considerata solo la prima metà del ciclo (t_c corrisponde al tempo di campionamento).

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.

L'errore di posizione medio è dato dunque dalla formula:

$$\epsilon = \frac{1}{N} \sum_{k=1}^N \frac{err_x(t)}{R} \quad (4.6)$$

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

4.1.4 Errore medio di posizione in y (emy)

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

$$\epsilon = \frac{1}{N} \sum_{k=1}^N \frac{y_s(t) - y_t(t)}{R} \quad (4.7)$$

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:

$$\epsilon = \frac{1}{R} \sqrt{\frac{\sum_{k=1}^N (x_s(t) - x_t(t))^2}{N}} \quad (4.8)$$

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:

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

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:

$$\epsilon = \frac{1}{N} \sum_{k=1}^N \frac{(err_x(t))(sign(err_x(t)) > 0)}{R} \quad (4.10)$$

dove err_x indica la quantità dell'equazione 4.4. 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:

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

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

4.1.9 Rox (ratio on x)

Il rox è dato dal rapporto tra la differenza dell'errore in x (emx) e lag error sulla differenza tra lead error e lag error. In formule risulta:

$$rox = \frac{emx(t) - lag(t)}{lead(t) - lag(t)} \quad (4.12)$$

Tale parametro indica di quanto il valor medio è spostato più verso il lead error (soggetto anticipa il target) o più verso il lag error (soggetto posticipa il target). È un valore percentuale, per cui un risultato pari al 100% indica un

totale spostamento verso il lead, mentre un valore pari allo 0% indica l'opposto spostamento verso il lag error.

4.1.10 τ – costante di tempo

Il parametro τ è stato realizzato per analizzare il tempo di adattamento del soggetto alla nuova situazione dopo il cambiamento di scala avvenuto durante il test, cioè il suo tempo di ritorno ad un valore di performance stabile. Il grafico 4.2 mostra in modo chiaro l'esponenziale decrescente della legge che descrive in modo completo l'andamento, in cui il parametro in questione rappresenta il tempo di discesa.

La legge che descrive tale comportamento è, in formule:

$$y = c + ke^{-\frac{x-x(1)}{\tau}} \quad (4.13)$$

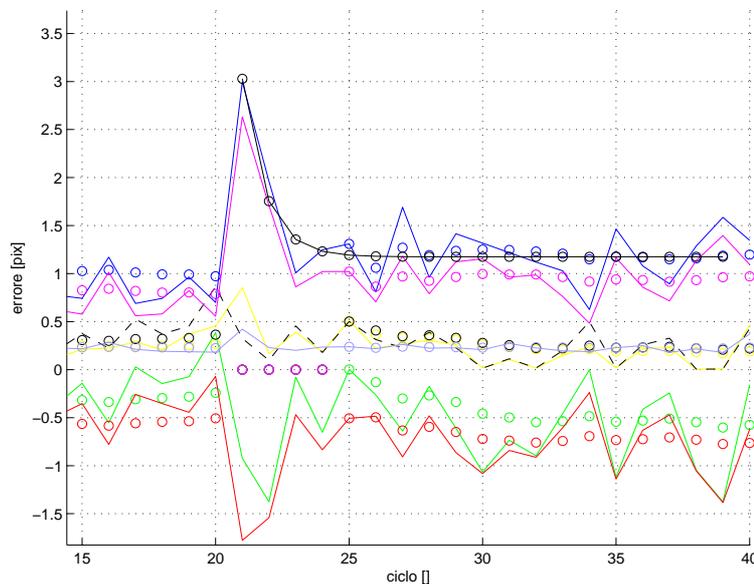


Figura 4.2: Grafico in cui si evidenzia l'andamento esponenziale dell'adattamento che si verifica subito dopo la comparsa della variazione di scala. Il tempo tau quantifica la durata della decrescita di tale legge, che nel grafico è segnalata da una linea nera.

dove:

- c = valore della funzione a tempo infinito
- k = valore che sommato a c fornisce il valore iniziale della funzione
- x = numero del ciclo considerato
- $x(1)$ = primo ciclo considerato, il 25° per il Pre_Test e il 21° per il Post_Test
- τ = costante di tempo dell'esponenziale (in termini di numero di cicli)

Tre dei parametri appena descritti (c , k , τ) non sono impostati a priori, ma vengono calcolati dalla funzione *fminsearch* con il criterio dei minimi quadrati. L'unico però ad essere preso in considerazione per l'analisi è τ , che infatti è il solo parametro che viene utilizzato nei file Excel durante l'elaborazione. Il codice realizzato in matlab permette di ottenere la legge a partire dai dati posseduti, tramite l'utilizzo della funzione di interpolazione passata alla funzione di mean search:

```
eval(['yy = datac.' varNames{i} '(cnum);'])
tau0 = [5 yy(end) yy(1)-yy(end)];
tau = fminsearch(@interpExp,tau0,[],xx,yy);
```

La funzione *interpExp* ha lo scopo di calcolare la distanza effettiva tra i dati reali e la possibile retta che li interpola, ottenuta dall'applicazione della funzione *funExp* che corrisponde alla 4.13; il risultato che produce (dist) è l'insieme delle distanze calcolate:

```
function dist = interpExp(tau,x,y)
    dist = sum((y-funExp(tau,x)).^2);
```

La matrice *dist* viene poi utilizzata dalla funzione *fminsearch*, che minimizza tali distanze, e quindi ci permette di ottenere la legge che al meglio rappresenta l'andamento decrescente.

Per ognuno dei parametri descritti sopra si calcolano media e deviazione standard:

- la media per ogni task è la media aritmetica delle medie relative ai singoli tratti

- la deviazione standard per ogni task è la media aritmetica delle deviazioni standard relative ai singoli tratti tranne che per i parametri dtt, eqmx, eqmy per cui si calcola la deviazione standard dei valori medi dei singoli cicli.

4.2 Test per l'analisi statistica

I test che vengono utilizzati per un'analisi statistica si dividono in test parametrici e non. Sono metodi statistici parametrici tutti quelli che si basano su distribuzioni probabilistiche quali la distribuzione normale o la distribuzione t di student. In contrapposizione, le tecniche non parametriche sono quelle che non sono basate su alcuna distribuzione probabilistica, quindi, in generale, i test non parametrici dovrebbero essere preferiti quando i dati non si distribuiscono secondo una normale, o comunque non si è in grado di dimostrarlo, ad esempio per numerosità ridotta dei campioni. Molti metodi non-parametrici si basano solo sull'ordine di grandezza dei dati, cioè sulla semplice graduatoria dei valori, trascurando i valori stessi. E' in tal modo che la statistica non-parametrica opera: liberandosi dai condizionamenti della distribuzione dei dati, ma al tempo stesso rinunciando ad ottenere da essa ogni possibile informazione.

Nel test *Variable Amplitude* i soggetti hanno eseguito tutti la medesima sequenza di task, però con diversi feedback audio: per questo motivo, nella maggior parte dei confronti utilizzeremo analisi per dati non appaiati, come il test di Mann-Whitney, l'Unpaired t-test e l'Unpaired t-test con Welch's correction, mentre per altri tipi di confronti useremo analisi per dati appaiati, come il test Wilcoxon e il t test. Il limite di significatività del p-value è stato impostato a 0,05.

4.2.1 Test per dati non appaiati

In molti casi non è fattibile o conveniente formare campioni dipendenti tra loro, poiché non si possono misurare gli effetti di differenti trattamenti sugli stessi individui: per questo motivo si sceglie di utilizzare test per dati non appaiati, in modo da poter analizzare in modo indipendente ogni gruppo e poi confrontarli tra loro.

Nel nostro caso abbiamo realizzato proprio questa attività, cioè abbiamo confrontato tra loro i risultati dei 4 gruppi a disposizione (NF, APER, ATRV, ATRJ) utilizzando 3 tipologie di test: *Mann-Whitney*, *Unpaired t-test* e *Unpaired t-test con Welch's correction*.

Test di Mann-Whitney

Il test di Mann-Whitney, detto anche la 'rank sum test', è un test non parametrico che mette a confronto gruppi di dati non appaiati. Per eseguire il test di Mann-Whitney, si ordinano in primo luogo tutti i valori dal più basso al più alto, senza prestare attenzione a quale gruppo appartiene ogni valore. Se due valori sono uguali, allora entrambi ottengono la media dei due ranghi per la quali si legano. Il valore più piccolo ottiene un rango pari a 1. Il valore maggiore ottiene un rango pari a di N, dove N è il numero totale di valori nei gruppi. Si sommano poi i ranghi in ciascun gruppo, e vengono segnalati i valori delle somme. Se le somme dei ranghi sono molto diverse, il valore di P (p-value) sarà piccolo.

Il valore del p-value risponde ad una determinata domanda: 'Se le popolazioni hanno realmente la stessa media, qual è la probabilità che un campionamento casuale porti ad una somma dei ranghi più lontana (o anche di più), come osservato in questo esperimento?' Se i campioni sono piccoli, e non ci sono legami di dipendenza tra loro, viene calcolato un valore esatto di P. Se invece i campioni sono grandi, o se ci sono legami di dipendenza, si approssima il valore di P tramite un'approssimazione gaussiana. In questo caso, il termine gaussiana ha a che fare con la distribuzione della somma dei ranghi, e non implica che i dati devono seguire una distribuzione gaussiana. L'approssimazione è abbastanza precisa con grandi campioni, ed è standard (utilizzata da tutti i programmi di statistica).

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 casualmente valori di medie distanti.

Unpaired t-test con Welch's correction

Nei t-test per dati non appaiati 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.

4.2.2 Test per dati appaiati

In generale, la caratteristica distintiva dei test per dati appaiati è la possibilità di accoppiare ogni osservazione di un campione con una e una sola osservazione dell'altro campione, con la necessità che i due gruppi abbiano sempre lo stesso numero di dati. Lo scopo principale dell'appaiamento dei dati è creare il massimo

di omogeneità entro ogni coppia, ma anche il massimo di eterogeneità tra le coppie.

Nel nostro caso, dal momento che tutti i soggetti hanno eseguito la stessa sequenza, si è scelto di confrontare anche la performance di ogni soggetto in punti diversi della procedura, ad esempio tra inizio del Pre_Test e parte finale dello stesso Pre_Test, e quindi è stato possibile effettuare un'analisi per dati appaiati. Per questa tipologia di analisi sono stati scelti il *Wilcoxon test* e il *t test*.

Wilcoxon test

Il Wilcoxon test è un test non parametrico che confronta la mediana di una colonna di numeri con una mediana ipotetica.

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?

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, dove con il termine 'errore' si fa riferimento alla differenza tra ogni valore e la mediana. Il risultato del Wilcoxon test ha 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.

t test

Il t test compara la media di una singola colonna di numeri con una media ipotetica inserita.

Il p-value risponde a questa domanda: se i dati sono prelevati da una popolazione gaussiana con una media equivalente all'ipotetico valore d'ingresso, qual è la probabilità di selezionare casualmente N campioni e trovare una media più lontana possibile dal valore ipotetico?

Se il p-value è elevato, dai dati non si può affermare che la media della popolazione sia diversa dal valore ipotetico immesso. Ciò non significa affermare che la media reale sia uguale al valore ipotetico, non si ha semplicemente una sostanziale differenza. Se il p-value è piccolo (di solito minore di 0,05), allora è improbabile che la differenza osservata tra la media osservata e quella ipotetica sia una coincidenza derivante dal campionamento casuale. Si può smentire l'idea per cui la differenza sia una coincidenza e concludere, invece, che la popolazione ha una media diversa rispetto al valore ipotetico immesso, quindi la differenza è statisticamente significativa.

Il t test assume che i dati siano stati campionati da una popolazione che segue una distribuzione gaussiana. Sebbene questa premessa non sia troppo importante quando si hanno campioni grandi, è importante con campioni di piccole dimensioni, soprattutto quando N è inferiore a 10. Se i dati non provengono da una distribuzione gaussiana ci sono tre opzioni. La scelta migliore è quella di modificare i valori per rendere la distribuzione più gaussiana, magari trasformando tutti i valori nei rispettivi reciproci o logaritmi. Un'altra possibilità è quella di utilizzare il Wilcoxon test non parametrico al posto del t test. Un'ultima opzione è quella di utilizzare comunque il t test, sapendo che il t test è abbastanza robusto per partenze da una distribuzione gaussiana con grandi campioni. Il t test presuppone anche che gli 'errori' siano indipendenti. Il termine 'errore' si riferisce alla differenza tra ogni valore e il gruppo di media. I risultati del t test acquistano significato solo quando la dispersione è casuale, cioè che un fattore che causa un valore troppo alto o troppo basso interessi solo per quel dato valore.

Informazioni relative alle varie analisi: outlier e range dei cicli analizzati

	NF	APER	ATRV	ATRJ	RANGE CICLI
pre-direct	0	4	2	3	25 / 10 - 24
pre-pin	0	1	1	0	10 - 24
pre-pfin	0	1	1	0	35 - 49
pre-deltap	0	0	2	0	35 - 49 / 10 - 24
pre-deltap-app	0	1	1	0	10 - 24 / 35 - 49
pre-tau	1	3	0	0	25 - 49
post-after	0	0	0	0	21 / 5 - 20
post-pin	0	1	1	0	5 - 20
post-pfin	0	1	1	0	25 - 39
post-deltap	1	3	1	0	25 - 39 / 5 - 20
post-deltap-app	0	1	1	0	5 - 20 / 25 - 39
post-tau	0	2	0	1	21 - 39
learning	0	1	1	0	10 - 24 (pre) / 25 - 39 (post)
retention	0	1	1	0	35 - 49 (pre) / 5 - 20 (post)

Tabella 4.1: Tabella con indicati i soggetti outlier di ogni gruppo, relativi ad ogni analisi effettuata, e i cicli presi in considerazione per tali gruppi di analisi.

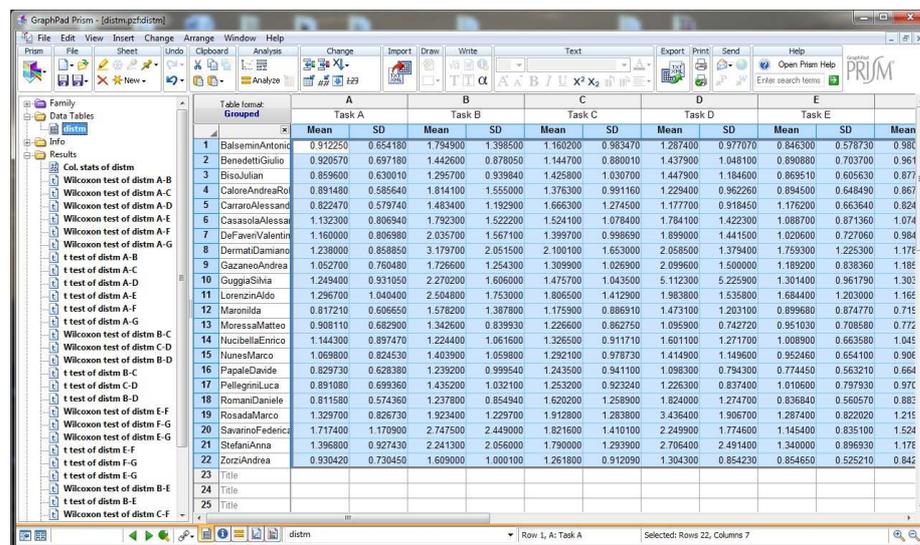
4.3 Elaborazioni dei dati

Per capire l'andamento generale dei soggetti, sono stati analizzati i grafici relativi ad ognuno dei parametri descritti sopra che evidenziavano il comportamento di tutti i soggetti. In tal modo è stato possibile individuare i valori che si discostavano maggiormente dalla media, così da capire quali soggetti eventualmente eliminare dall'analisi per non alterare i dati. In base a questa valutazione, dei soggetti totali che hanno partecipato ne sono stati scartati diversi, a seconda del tipo di analisi effettuata. L'operazione successiva è stata quella di individuare il range di cicli entro il quale effettuare l'analisi. E' palese, infatti, che nei primi cicli la performance sarà irregolare perché il soggetto non ha preso ancora dimestichezza con il joystick, mentre negli ultimi si può verificare un certo affaticamento (vedi 4.1).

Sempre a partire da tali grafici è stato possibile individuare a colpo d'occhio comportamenti anomali del soggetto, dovuti a possibili distrazioni, così da capire quali cicli dovranno essere eventualmente scartati. Questa operazione consiste nel sostituire il ciclo incriminato con il precedente, in modo da eliminare le irregolarità.

4.3.1 GraphPad Prism

Per le conclusioni statistiche sui dati memorizzati si è fatto ricorso al software GraphPad Prism: si tratta di 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 funzioni statistiche in grado di aggiornarsi istantaneamente nel momento in cui si vanno a cambiare i dati dalla tabella di origine. Sono stati generati un numero di file.pzf pari alle relative misure calcolate, per esempio nell'analisi della fase Variabile sono state considerate 8 misure e quindi generati 8 file.pzf. All'interno del programma sono stati impostati tutti i confronti desiderati, da effettuare con la totalità di test di analisi precedentemente evidenziati (Mann-Whitney, Wilcoxon ecc); ogni file.pzf importa il file della tabella Excel del parametro corrispondente generando così un collegamento. Ogni file restituisce un istogramma della media e deviazione standard di ogni task per la corrispondente misura e il p-value relativo ad ogni confronto effettuato.



	Task A		Task B		Task C		Task D		Task E		
	Mean	SD									
1 BaliseimAntonic	0.912250	0.654180	1.794900	1.398500	1.160200	0.983470	1.287400	0.977070	0.846300	0.578730	0.980
2 BenedettiGiulio	0.920570	0.697180	1.442600	0.878050	1.144700	0.880010	1.437900	1.048100	0.890880	0.703700	0.961
3 BisouJulian	0.859600	0.630010	1.295700	0.939840	1.425800	1.030700	1.447900	1.184600	0.869510	0.605630	0.877
4 CaloreAndreaRob	0.891480	0.896640	1.914100	1.555000	1.376300	0.991160	1.229400	0.922260	0.894500	0.548490	0.867
5 CantaraAlessand	0.822470	0.579740	1.483400	1.192900	1.666300	1.274500	1.177700	0.918450	1.176200	0.663540	0.824
6 CasasolaAlessa	1.132300	0.806940	1.792300	1.522200	1.524100	1.078400	1.784100	1.422300	1.068700	0.871360	1.074
7 DaFaveriValentin	1.160000	0.806980	2.035700	1.567100	1.399700	0.998950	1.899000	1.441500	1.020600	0.727060	0.984
8 DermatDamiano	1.238000	0.858850	3.179700	2.051500	2.100100	1.653000	2.056500	1.379400	1.759300	1.225300	1.176
9 GazaneoAndrea	1.052700	0.760480	1.726600	1.254300	1.309900	1.026900	2.099600	1.500000	1.189200	0.838360	1.188
10 GuggiaSilvia	1.249400	0.931050	2.270200	1.606000	1.475700	1.043500	5.112300	5.225900	1.301400	0.961790	1.303
11 LorenzinAldo	1.296700	1.040400	2.504800	1.753000	1.806500	1.412900	1.983800	1.535800	1.684400	1.203000	1.166
12 Maronilda	0.817210	0.606650	1.578200	1.387800	1.175900	0.886910	1.473100	1.203100	0.899680	0.874770	0.715
13 MoressaMatteo	0.908110	0.682900	1.342600	0.839930	1.226600	0.862750	1.095900	0.742720	0.951030	0.708680	0.772
14 NucibellaEnrico	1.144300	0.897470	1.224400	1.061600	1.326500	0.911710	1.601100	1.271700	1.008900	0.663580	1.045
15 NunesMarco	1.069800	0.824530	1.403900	1.059800	1.292100	0.978730	1.414900	1.149600	0.952460	0.654100	0.906
16 PapaleDavide	0.829730	0.628380	1.239200	0.999540	1.243500	0.941100	1.096300	0.794300	0.774450	0.563210	0.664
17 PellagniniLuca	0.891080	0.699360	1.435200	1.032180	1.253200	0.923240	1.226300	0.837400	1.010600	0.797930	0.977
18 RomaniDanielle	0.811580	0.574360	1.237800	0.854940	1.620200	1.258900	1.824000	1.274700	0.836840	0.560570	0.888
19 RosadaMarco	1.329700	0.826730	1.923400	1.229700	1.912800	1.283800	3.436400	1.906700	1.287400	0.822020	1.215
20 SavarinoFederica	1.771400	1.170900	2.747500	2.449000	1.821600	1.410100	2.249900	1.774600	1.145400	0.835100	1.524
21 Stefanikena	1.396800	0.927430	2.241300	2.056000	1.790000	1.293900	2.706400	2.491400	1.340000	0.896930	1.176
22 ZorziAndrea	0.930420	0.739045	1.609000	1.000100	1.261800	0.912090	1.304300	0.854230	0.854650	0.525210	0.842
23 Title											
24 Title											
25 Title											

Figura 4.3: Prism: software utilizzato per l'analisi statistica

4.3.2 Procedura

Come spiegato nel precedente capitolo, i dati relativi ad ogni task eseguito da ciascun soggetto sono stati salvati nella struttura *ex.data*, che contiene tutti i parametri necessari per un'analisi offline. Su tale struttura, innanzitutto, sono state effettuate una serie di operazioni per predisporre i dati in una forma più adatta per l'elaborazione. Sono stati utilizzati i seguenti script:

1. *saveDat.m*: carica i file *.mat* ottenuti dall'esecuzione dei test e calcola i parametri utili ad una prima elaborazione, salvando i risultati in un corrispondente file *.dat*. In particolare per ogni fase del test, vengono calcolati tutti i parametri definiti nel paragrafo precedente: errore di posizione, integrale del modulo della velocità (dtt), ritardo medio (errore di posizione lungo l'asse x - emx), errore di posizione lungo l'asse y (emy), errore quadratico medio in x (eqmx), errore quadratico medio in y (eqmy), lead error e lag error. Tali misure vengono effettuate, una prima volta, lungo una finestra di osservazione pari alla durata totale dell'esercizio, e poi vengono mediate ciclo per ciclo; per ogni misura viene calcolata sia la media che la deviazione standard; i valori ottenuti sono infine salvati nelle corrispondenti strutture *data* e *datac*
2. *saveDatTaroc.m*: elimina da alcuni soggetti i cicli anomali, ovvero quei cicli isolati rispetto la prestazione dell'intero esercizio, dovuti probabilmente ad istanti di distrazione del soggetto. Come spiegato sopra, i cicli anomali non vengono eliminati, bensì si è scelto di copiare sui dati del ciclo incriminato quelli del suo ciclo precedente, in modo da eliminare le irregolarità ed avere una caratteristica più omogenea. All'interno del file bisogna specificare il nome del soggetto, il task e il numero dei cicli da sovrascrivere. Le strutture *data* e *datac* vengono quindi aggiornate
3. *xlsDatAmplitude_tipoAnalisi.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 (cioè media e deviazione standard) relative a ciascun parametro considerato (come emx, emy, dtt ecc).

Una volta eseguiti tali calcoli, salva i dati ottenuti su file `.xls`, all'interno della cartella *Risultati-amplitude/tipoAnalisi*, utili ad un successivo utilizzo in Prism. Viene inoltre generato il file `settings.mat` per tenere traccia dei parametri impostati al lancio del file

4. *pScript.pzc*: implementato in Prism, apre ogni file `.pzf` contenuto nella cartella, estrae i valori dei p-value e li copia per l'esportazione nei seguenti file `.csv`:
 - file `MW.csv` (confronti con Mann-Whitney), `UW.csv` (Unpaired t-test) e `U.csv` (Unpaired-test con Welch's corrections) per tutte le analisi con dati non appaiati (vedi sez. 4.5, 4.6 e 4.4).
 - file `Wilcoxon.csv` e `t.test.csv` per le analisi con dati appaiati (vedi sez. 4.5, 4.6 e 4.7).
5. a questo punto si aprono in contemporanea i file `.csv` e il file *p-values-amplitude.xlsx*, che contiene tanti fogli quanti sono i file `.csv`. Il file `p-values-amplitude.xlsx` viene aggiornato automaticamente prendendo i contenuti dai vari `.csv`, pertanto è necessario salvarlo prima di ogni chiusura.
6. *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 Variabile

In questo paragrafo vengono presentati i principali risultati ottenuti dall'elaborazione dei dati riguardanti la fase del test denominata Variabile.

Sono state confrontate tutte le grandezze definite in 4.1 tra le seguenti coppie di task:

- NF - APER
- NF - ATRV
- NF - ATRJ
- APER - ATRV
- APER - ATRJ
- ATRV - ATRJ

In base alle considerazioni fatte durante lo studio dei risultati delle analisi, è possibile esprimere le conclusioni sul lavoro relativo alla fase *Variabile* 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 (APER) presentano, in termini di accuratezza (*distm*), esiti meno soddisfacenti rispetto ai soggetti che appartenevano rispettivamente ai gruppi privi di audio (NF) o con audio proporzionale alla velocità del target (sia con riferimento alla scala video che scala joystick).
- L'insieme formato da feedback audio legato alla velocità del target con scalatura a video (ATRV) e feedback audio legato alla velocità del target con scalatura rispetto al joystick (ATRJ), risultano essere dal punto di vista dell'accuratezza (*distm*) di un più valido aiuto rispetto all'audio legato all'errore di posizione.

- Il fatto che ATRV, a differenza di ATRJ, risulti significativamente migliore di NF in termini di performance (*emx*, test parametrici) suggerisce che il soggetto preferisca ricevere un feedback legato al task visualizzato piuttosto che legato al task motorio da eseguire.
- In termini di performance (*emx*), il gruppo di soggetti caratterizzati da feedback audio legato all'errore di posizione rispetto al target (APER) risultavano essere penalizzati rispetto agli altri gruppi. Si può ben notare dall'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.

Di seguito vengono riportati i grafici relativi (4.4 e 4.5)¹.

¹Per esportare i grafici in EPS direttamente da Prism è necessario impostare, nel campo Export Graph del menù: 'Color mode → Grayscale', 'Embed fonts in this file' → No (More compatible).

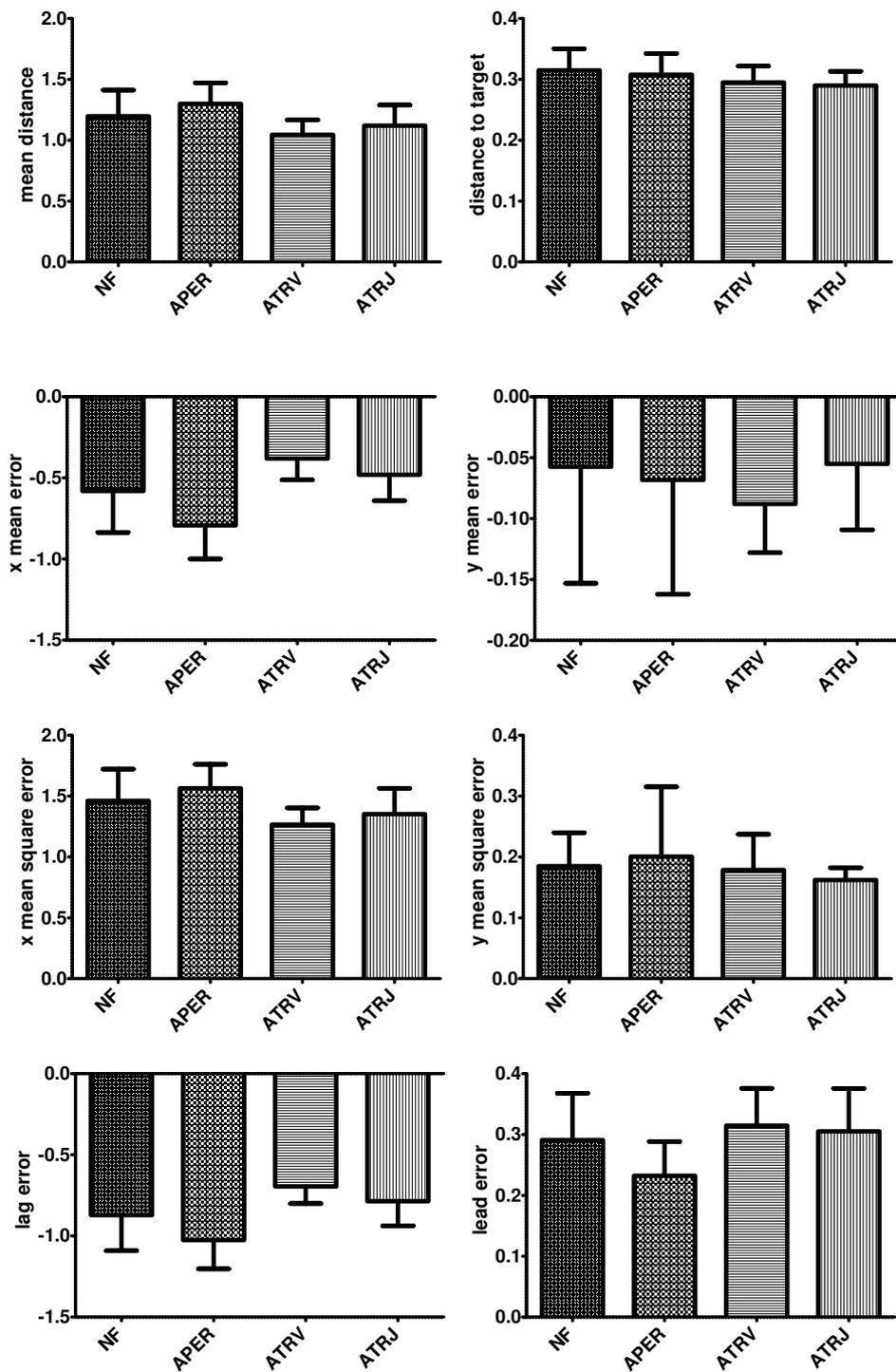


Figura 4.4: Grafici relativi ai parametri considerati nell'elaborazione: distm, dtt, emx, eqmx, lag, lead. Tutte le grandezze sono state normalizzate rispetto al raggio del target tranne il dtt, normalizzato rispetto alla lunghezza del tratto percorso.

P-VALUE var (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

Figura 4.5: Tabella dei p-value relativa alle analisi per il test Variabile

4.5 Risultati Pre_Test

4.5.1 Pre-Direct

L'analisi denominata 'Direct' ha lo scopo principale di visualizzare l'effetto diretto di una variazione avvenuta durante il task, e prende in considerazione il rapporto tra il comportamento tenuto dal soggetto nel ciclo di variazione della scala (cioè il 25° ciclo) rispetto all'andamento medio nei cicli precedenti (cioè il range che va dal 10° al 24° ciclo).

Da quest'analisi emerge che:

- relativamente al parametro *distm*, si nota come l'effetto diretto sia visibile per tutti i soggetti sottoposti al test. Da ciò si può concludere che si verifica una perdita di accuratezza nell'esecuzione del task. Tale perdita è significativamente inferiore per il gruppo APER, a indicare che un feedback legato all'errore aiuta a contenerne l'entità durante la variazione improvvisa della scala. Il dato è confermato dal parametro *eqmx*.
- il parametro *rox* evidenzia come nel gruppo ATRV la perdita di performance (salto in avanti) sia più contenuta rispetto ai gruppi NF ed APER. Anche ATRJ nei test parametrici risulta significativamente inferiore all'APER. Questo risultato sta ad indicare che un feedback legato al task aiuta a contenere la perdita di performance durante la variazione. Il fatto che ATRV sia lievemente migliore di ATRJ suggerisce come il soggetto durante la variazione faccia maggiore affidamento sulla modalità visiva (nel gruppo ATRJ infatti il suono è coerente col movimento richiesto e non con quello visualizzato).

4.5.2 Pre-Pin

L'analisi denominata 'Pre-Pin' ha lo scopo principale di visualizzare l'andamento medio dei parametri nei primi cicli (dal 10° al 24°) del Pre_Test e visualizzare il comportamento tenuto dal soggetto in questo primo approccio con il task.

Da quest'analisi emerge che:

- soffermandosi sul parametro emx , il gruppo con audio legato alla velocità del target visualizzato(ATRV) ha una performance mediamente migliore rispetto al gruppo con audio legato all'errore di posizione(APER).
- considerando l'accuratezza dell'inseguimento ($distm$), si nota come essa non dipenda dal tipo di feedback utilizzato.

Entrambe le osservazioni possono essere viste come una conferma delle conclusioni ottenute nel lavoro effettuato precedentemente [49] (vedi sez.3.1), dove attraverso altre tipologie di test, si ottenevano risultati del tutto compatibili con questi.

4.5.3 Pre-PFin

L'analisi denominata 'Pre-PFin' ha lo scopo principale di visualizzare l'andamento medio dei parametri nei cicli successivi al cambio scala (dal 39° al 45°) del Pre_Test e visualizzare il comportamento tenuto dal soggetto in questo primo approccio con la nuova tipologia di scala realizzata.

I grafici confermano i risultati ottenuti con l'analisi denominata 'Pre-PIn', anche se per il parametro emx non si raggiunge la significatività statistica a causa del maggior valore delle varianze (presumibilmente dovuto all'avvenuto cambio scala).

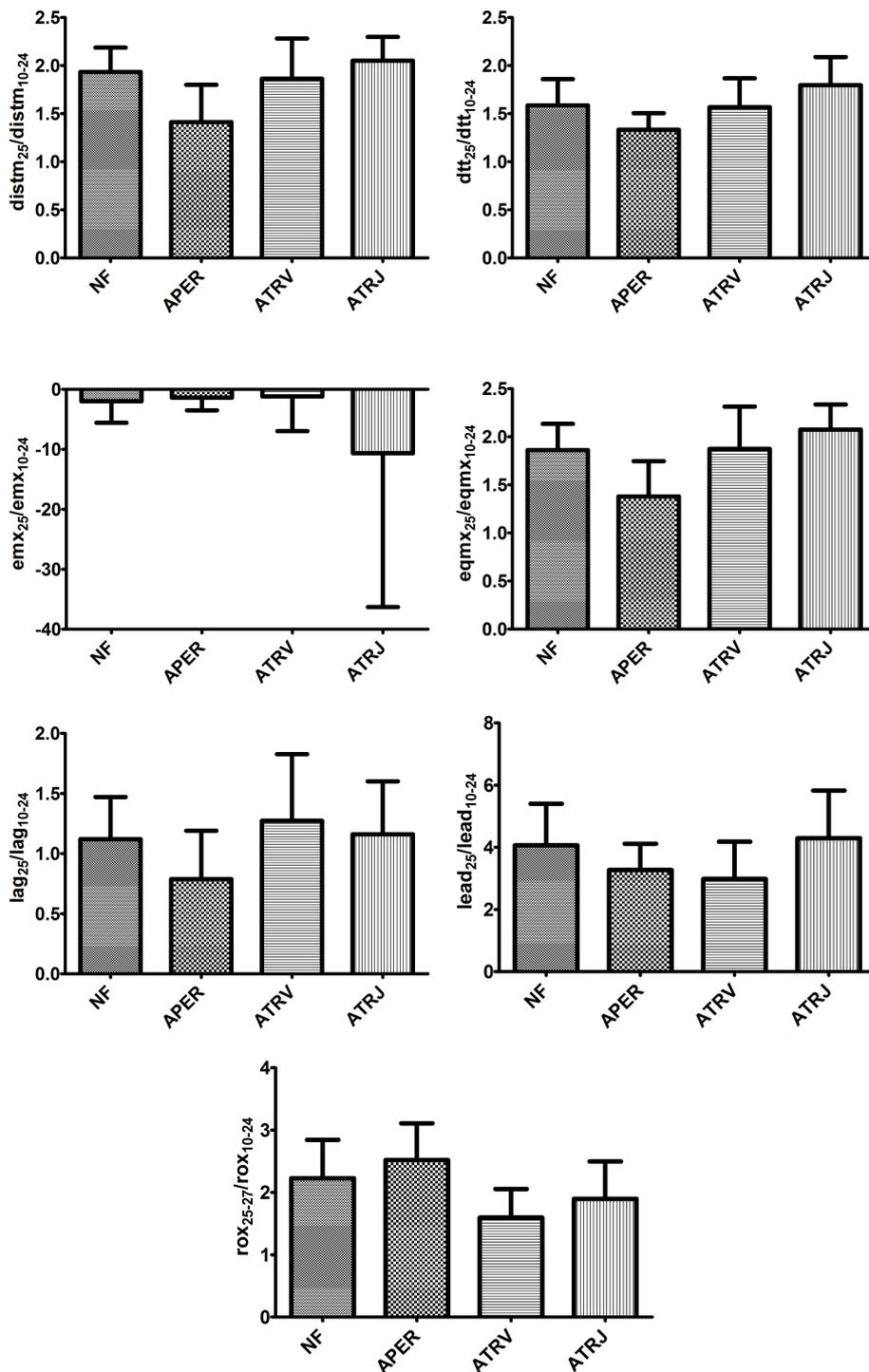


Figura 4.6: Grafici relativi all'analisi Pre-Direct del Pre-Test, con parametri considerati: $distm$, dtt , emx , $eqmx$, lag , $lead$ e rox . Tutte le grandezze sono state normalizzate rispetto al raggio del target tranne il dtt , normalizzato rispetto alla lunghezza del tratto percorso.

P-VALUE pre-direct (Mann Whitney)						
	NF-APER	NF-ATRV	NF-ATRJ	APER-ATRV	APER-ATRJ	ATRV-ATRJ
distm	0.0093	0.5035	0.2545	0.0434	0.0010	0.2110
dtm	0.0523	0.5974	0.1489	0.0343	0.0006	0.0435
emx	0.9671	0.4597	0.4033	0.5726	0.3704	0.4002
eqmx	0.0118	0.8053	0.1489	0.0266	0.0006	0.4002
lag	0.0632	0.9719		0.0205	0.0592	0.9048
lead	0.1731	0.0620		0.4082	0.2359	0.0653
rox	0.4328	0.0265	0.2242	0.0031	0.0592	0.2775

P-VALUE pre-direct (Unpaired t-test)						
	NF-APER	NF-ATRV	NF-ATRJ	APER-ATRV	APER-ATRJ	ATRV-ATRJ
distm	0.0024	0.6437	0.3105	0.0319	0.0009	0.2575
dtm	0.0355	0.8853	0.1124	0.0692	0.0014	0.1099
emx	0.6741	0.7072	0.2798	0.9341	0.3256	0.2708
eqmx	0.0042	0.9445	0.0920	0.0218	0.0004	0.2464
lag	0.0714	0.4574	0.8201	0.0549	0.0888	0.6368
lead	0.1589	0.0669	0.7238	0.5766	0.1148	0.0520
rox	0.3152	0.0159	0.2418	0.0018	0.0480	0.2313

P-VALUE pre-direct (Unpaired t-test with Welch's)						
	NF-APER	NF-ATRV	NF-ATRJ	APER-ATRV	APER-ATRJ	ATRV-ATRJ
distm	0.0068	0.6532	0.3102	0.0315	0.0021	0.2487
dtm	0.0259	0.8860	0.1168	0.0576	0.0014	0.1105
emx	0.6489	0.7147	0.3439	0.9284	0.3112	0.3107
eqmx	0.0084	0.9460	0.0916	0.0203	0.0008	0.2379
lag	0.0829	0.4701	0.8247	0.0484	0.0887	0.6330
lead	0.1332	0.0664	0.7281	0.5621	0.1091	0.0570
rox	0.3136	0.0151	0.2414	0.0030	0.0491	0.2414

Figura 4.7: Tabella dei p-value relativa alle analisi 'Direct' per il test Pre_Test

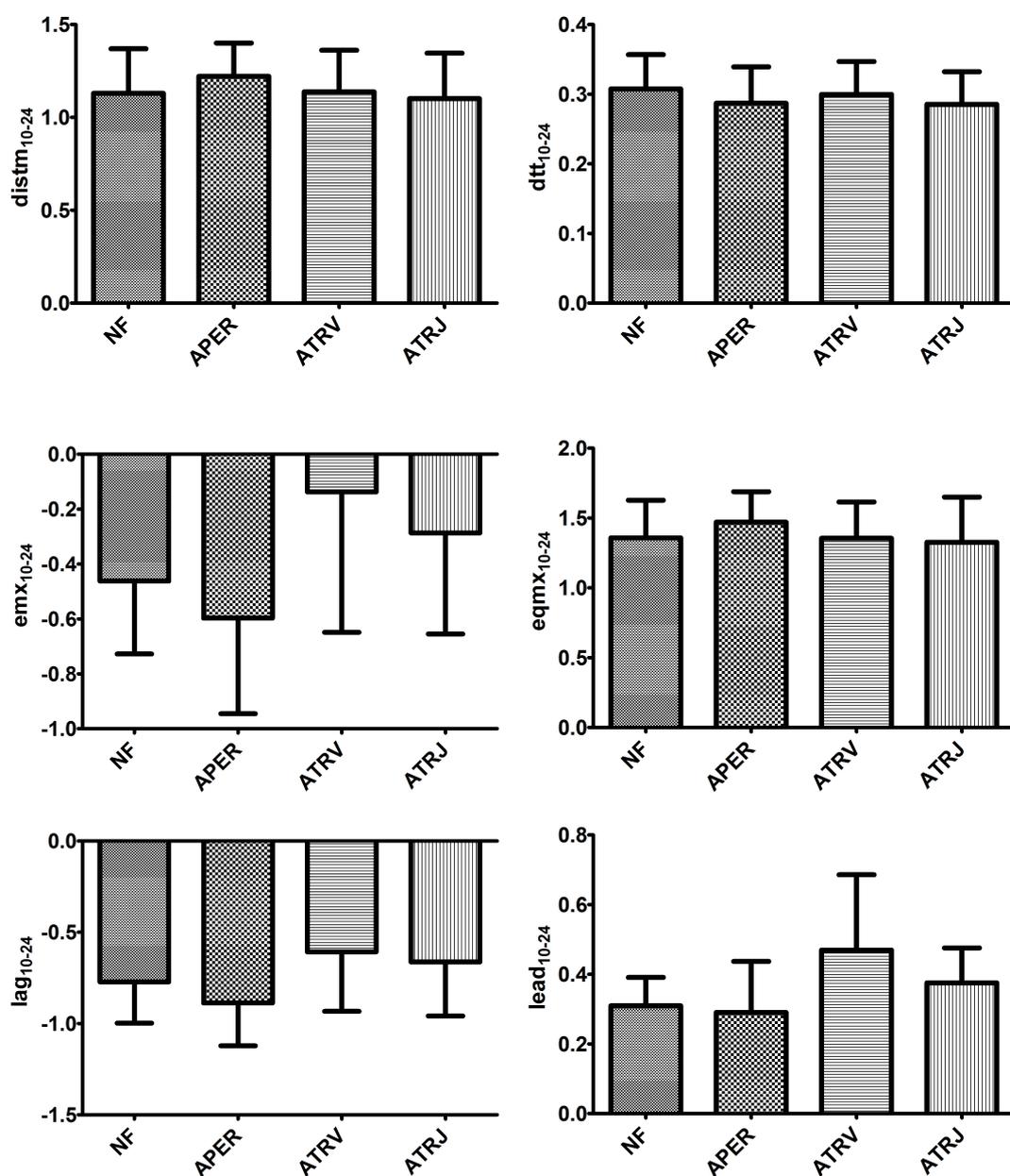


Figura 4.8: Grafici relativi all'analisi Pre-Pin del Pre-Test, con parametri considerati: $distm$, dtt , emx , $eqmx$, lag , $lead$. Tutte le grandezze sono state normalizzate rispetto al raggio del target tranne il dtt , normalizzato rispetto alla lunghezza del tratto percorso.

P-VALUE pre-pin (Mann Whitney)						
	NF-APER	NF-ATRV	NF-ATRJ	APER-ATRV	APER-ATRJ	ATRV-ATRJ
distm	0.2372	0.7928	0.8777	0.2934	0.2301	0.7818
dtm	0.5114	0.8955	0.5181	0.7427	0.9264	0.5181
emx	0.3246	0.0569	0.2301	0.0104	0.0605	0.4061
eqmx	0.2643		0.8294	0.3246	0.1481	0.6009
lag	0.2372	0.0660	0.2301	0.0418	0.0694	0.5588
lead	0.6936	0.0762	0.1481	0.0569	0.0906	0.2301

P-VALUE pre-pin (Unpaired t-test)						
	NF-APER	NF-ATRV	NF-ATRJ	APER-ATRV	APER-ATRJ	ATRV-ATRJ
var-distm	0.3222	0.9373	0.7784	0.3442	0.1959	0.7119
var-dtm	0.3529	0.7043	0.2807	0.5576	0.9390	0.4756
var-emx	0.3183	0.0764	0.2087	0.0229	0.0508	0.4271
var-eqmx	0.2949	0.9860	0.8060	0.2755	0.2299	0.8156
var-lag	0.2508	0.1846	0.3357	0.0313	0.0576	0.6730
var-lead	0.7110	0.0325	0.0993	0.0341	0.1161	0.1892

P-VALUE pre-pin (Unpaired t-test with Welch's)						
	NF-APER	NF-ATRV	NF-ATRJ	APER-ATRV	APER-ATRJ	ATRV-ATRJ
var-distm	0.3235	0.9374	0.7783	0.3448	0.1906	0.7110
var-dtm	0.3535	0.7045	0.2823	0.5580	0.9393	0.4761
var-emx	0.3197	0.0814	0.2039	0.0247	0.0509	0.4352
var-eqmx	0.2956	0.9861	0.8046	0.2762	0.2237	0.8139
var-lag	0.2515	0.1872	0.3306	0.0326	0.0559	0.6746
var-lead	0.7123	0.0404	0.0971	0.0362	0.1253	0.2104

Figura 4.9: Tabella dei p-value relativa alle analisi 'PIn' per il test Pre_Test

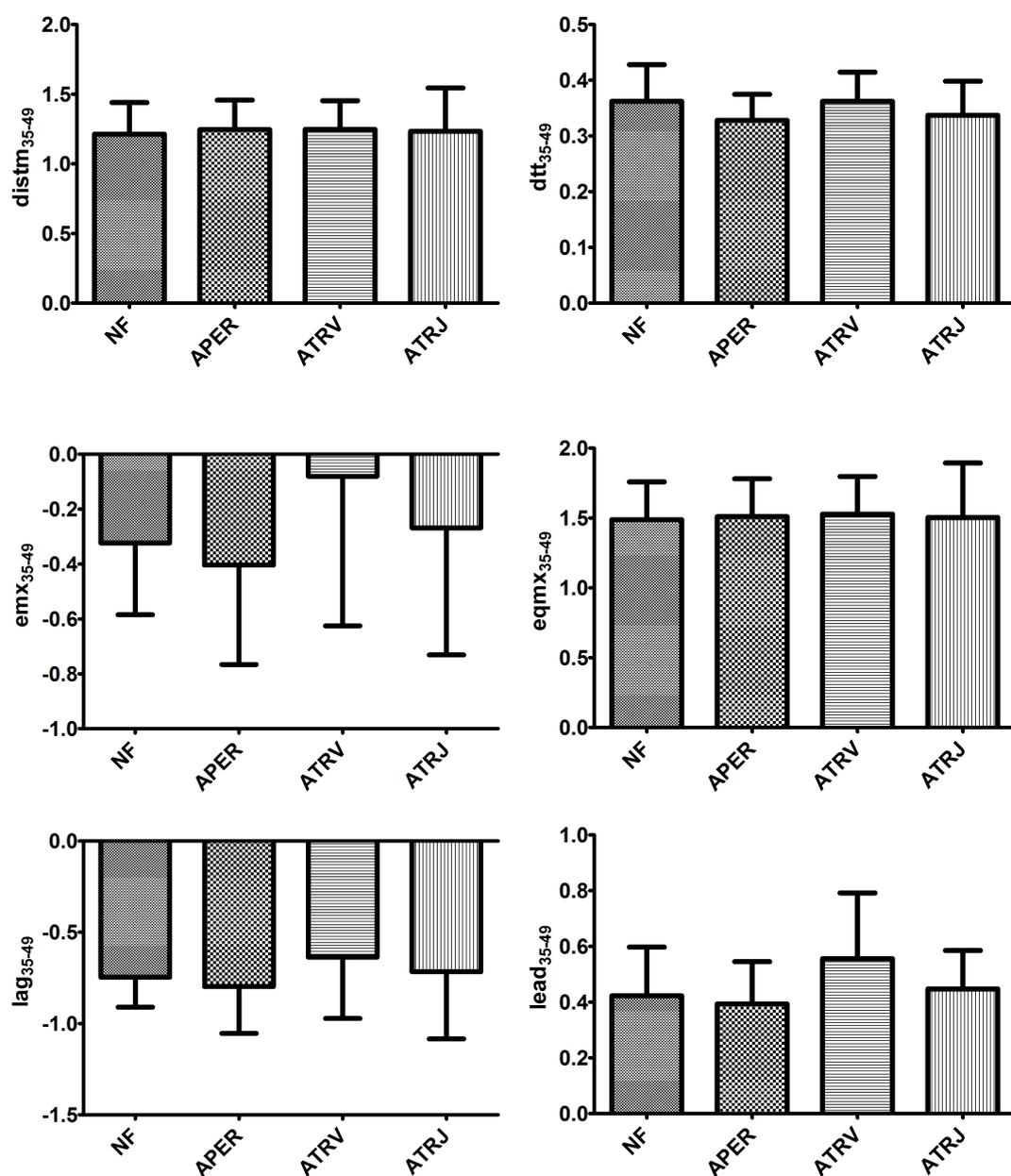


Figura 4.10: Grafici relativi all'analisi Pre-PFin del Pre-Test, con parametri considerati: $distm$, dtt , emx , $eqmx$, lag , $lead$. Tutte le grandezze sono state normalizzate rispetto al raggio del target tranne il dtt , normalizzato rispetto alla lunghezza del tratto percorso.

P-VALUE pre-pfin (Mann Whitney)						
	NF-APER	NF-ATRV	NF-ATRJ	APER-ATRV	APER-ATRJ	ATRV-ATRJ
distm	0.7427	0.5114	0.8777	0.9476	0.6009	0.6444
dtm	0.3246	0.9476	0.4061	0.2643	0.6444	0.4791
emx	0.7427	0.1486	0.2071	0.1007	0.1858	0.1858
eqmx		0.5994	0.8777		0.8777	0.6009
lag	0.8438	0.1679	0.1661	0.1310	0.2815	0.3722
lead	0.8438	0.1150	0.5588	0.0569	0.2815	0.2071

P-VALUE pre-pfin (Unpaired t-test)						
	NF-APER	NF-ATRV	NF-ATRJ	APER-ATRV	APER-ATRJ	ATRV-ATRJ
distm	0.7343	0.7118	0.8635	0.9795	0.9138	0.8967
dtm	0.1753	0.9986	0.3549	0.1224	0.6917	0.3065
emx	0.5552	0.1977	0.7345	0.1165	0.4462	0.3806
eqmx	0.8555	0.7338	0.9157	0.8747	0.9658	0.8637
lag	0.5815	0.3433	0.8080	0.2214	0.5497	0.5946
lead	0.6792	0.1477	0.7082	0.0687	0.3802	0.1877

P-VALUE pre-pfin (Unpaired t-test with Welch's)						
	NF-APER	NF-ATRV	NF-ATRJ	APER-ATRV	APER-ATRJ	ATRV-ATRJ
distm	0.7345	0.7120	0.8616	0.9795	0.9125	0.8949
dtm	0.1770	0.9986	0.3570	0.1232	0.6885	0.3038
emx	0.5559	0.2040	0.7293	0.1192	0.4417	0.3851
eqmx	0.8556	0.7340	0.9144	0.8748	0.9653	0.8617
lag	0.5825	0.3482	0.8032	0.2230	0.5441	0.5934
lead	0.6794	0.1494	0.7115	0.0712	0.3828	0.2028

Figura 4.11: Tabella dei p-value relativa alle analisi 'PFin' per il test Pre_Test

4.5.4 Pre-Deltap

L'analisi denominata 'Pre-Deltap' ha lo scopo principale di visualizzare la relazione tra le fasi del Pre_Test che precedono e seguono la variazione della scala rispetto alla prima fase del test stesso; essa prende in considerazione il rapporto dato dalla differenza del comportamento tenuto dal soggetto nell'ultima fase del test (cioè dal 35° al 49° ciclo) con la prima fase (cioè dal 10° al 24° ciclo) rispetto a quest'ultima, cioè all'andamento medio nei primi cicli effettuati.

Dalle analisi effettuate è emerso che:

- in termini di performance (*emx*) non si riscontrano differenze significative tra i diversi gruppi.
- in termini di accuratezza (*distm*, *dtm*) non si riscontrano differenze significative tra i diversi gruppi, tranne nel confronto APER - ATRJ che nei test parametrici risulta significativo a favore di APER (*distm*).

Possiamo concludere che le differenze di performance nelle due condizioni di prova (scala iniziale e scala finale) non dipendono dal particolare tipo di feedback utilizzato. Questo risultato conferma quanto osservato nel Pre-PFin, cioè che gli andamenti dei grafici erano simili a quelli del Pre-Pin: se una modalità di feedback prevale sulle altre, lo fa in tutte le condizioni di prova.

4.5.5 Pre-Deltap-App

L'analisi denominata 'Pre-Deltap-App' ha lo scopo principale di visualizzare la relazione tra la fase iniziale del Pre_Test (cioè dal 10° al 24° ciclo) e la parte finale della stessa fase (cioè dal 35° al 49° ciclo), cioè tra le due parti che presentano una diversa scala video all'interno della stessa fase di test. Lo scopo è quello di visualizzare quale relazione intercorre tra i cicli precedenti e quelli che seguono la variazione della scala nei vari soggetti (è stata infatti utilizzata un'analisi per dati appaiati [sez.4.2.2]).

Dalle analisi è emerso che:

- in tutti i gruppi c'è un aumento della distanza percorsa intorno al target (*dtm*) tra la fase iniziale e la fase finale, ad indicare come il rapporto di scala

1:1 consenta di avere maggiore accuratezza nei movimenti (da notare che tutte i parametri sono calcolati sulla scala del video, che nella fase finale è doppia rispetto alla scala del joystick).

- la performance (emx) è comparabile nella fase iniziale e nella fase finale in tutti i gruppi, tranne in APER, che presenta performance migliori nella fase finale. Questo risultato si può spiegare considerando che l'amplificazione di scala nella fase finale determina di fatto un raddoppio del guadagno del feedback sonoro rispetto alla scala di movimento del soggetto. In altre parole, uno stesso errore del soggetto viene tradotto in un suono più forte. Il fatto che l'incremento del feedback determini un miglioramento della performance suggerisce che tale feedback sia utilizzato direttamente dal soggetto nel controllo motorio.

Sarebbe interessante ripetere le analisi per dati appaiati, dopo aver raggruppati tutti i soggetti in un unico gruppo.

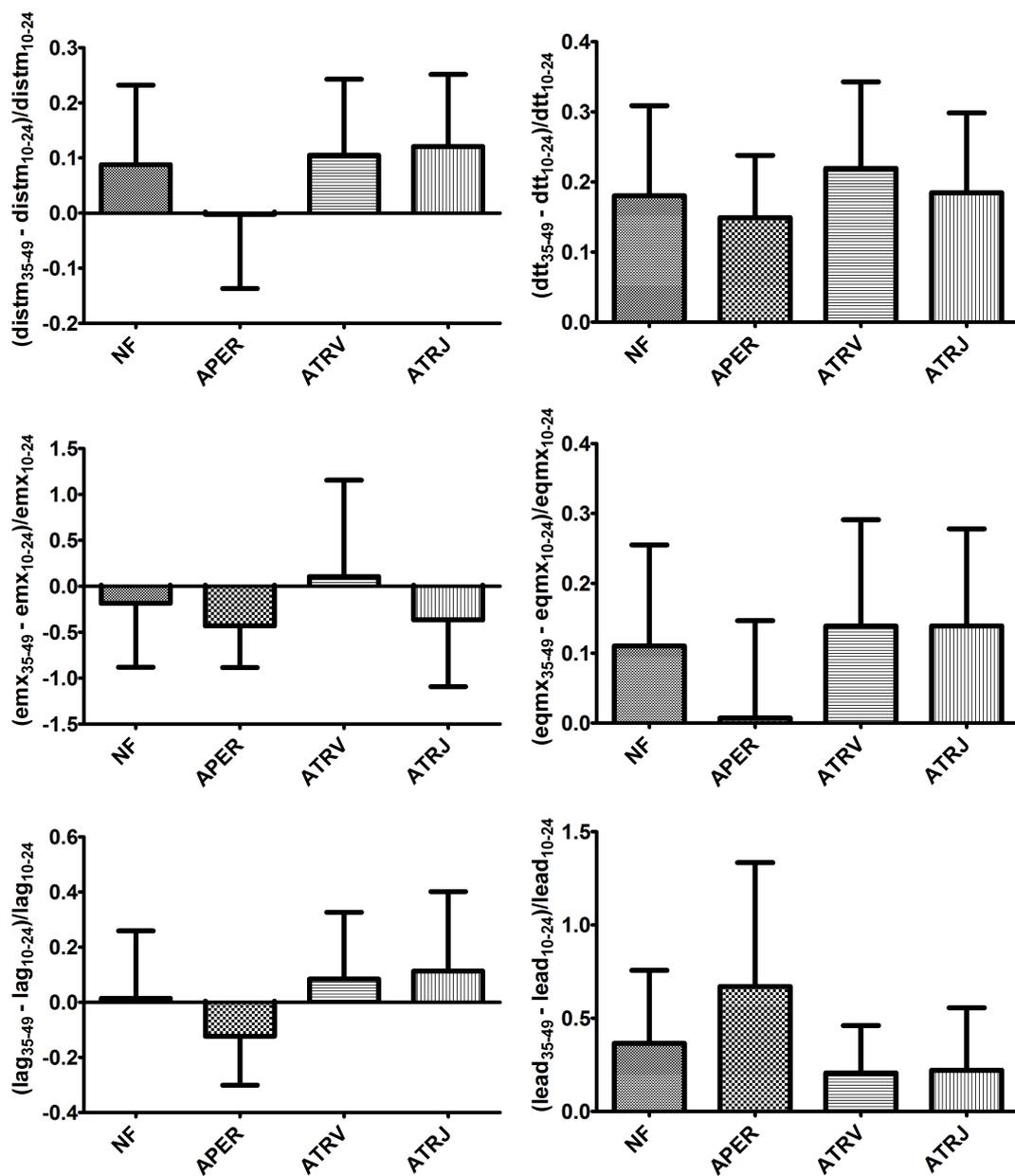


Figura 4.12: Grafici relativi all'analisi Pre-Deltap del Pre_Test, con parametri considerati: distm, dtt, emx, eqmx, lag, lead. Tutte le grandezze sono state normalizzate rispetto al raggio del target tranne il dtt, normalizzato rispetto alla lunghezza del tratto percorso.

P-VALUE pre-deltap (Mann Whitney)						
	NF-APER	NF-ATRV	NF-ATRJ	APER-ATRV	APER-ATRJ	ATRV-ATRJ
distm	0.0605	0.9159	0.6891	0.0602	0.0531	0.7667
dtm	0.3401	0.2751	0.7350	0.1062	0.3708	0.4887
emx	0.3722	0.5974	0.3722	0.1563	0.9310	0.1563
eqmx	0.0605	0.7513	0.8294	0.0443	0.0304	0.7667
lag	0.1166	0.5974	0.6891	0.0518	0.0606	0.9737
lead	0.5181	0.5495	0.5181	0.1379	0.0885	0.9737

P-VALUE pre-deltap (Unpaired t-test)						
	NF-APER	NF-ATRV	NF-ATRJ	APER-ATRV	APER-ATRJ	ATRV-ATRJ
distm	0.1371	0.7841	0.5687	0.0812	0.0331	0.7827
dtm	0.5040	0.4846	0.9332	0.1358	0.4055	0.4977
emx	0.3215	0.4601	0.5480	0.1240	0.7947	0.2283
eqmx	0.0962	0.6650	0.6332	0.0471	0.0300	0.9980
lag	0.1355	0.5183	0.3877	0.0311	0.0243	0.8044
lead	0.2001	0.2883	0.3513	0.0509	0.0483	0.9094

P-VALUE pre-deltap (Unpaired t-test with Welch's)						
	NF-APER	NF-ATRV	NF-ATRJ	APER-ATRV	APER-ATRJ	ATRV-ATRJ
distm	0.1391	0.7838	0.5707	0.0826	0.0336	0.7842
dtm	0.5125	0.4842	0.9336	0.1510	0.4064	0.5017
emx	0.3337	0.4711	0.5475	0.1608	0.7952	0.2482
eqmx	0.0974	0.6661	0.6340	0.0502	0.0305	0.9980
lag	0.1433	0.5185	0.3847	0.0386	0.0264	0.8014
lead	0.1930	0.2804	0.3558	0.0426	0.0529	0.9072

Figura 4.13: Tabella dei p-value relativa alle analisi 'Deltap' per il test Pre_Test

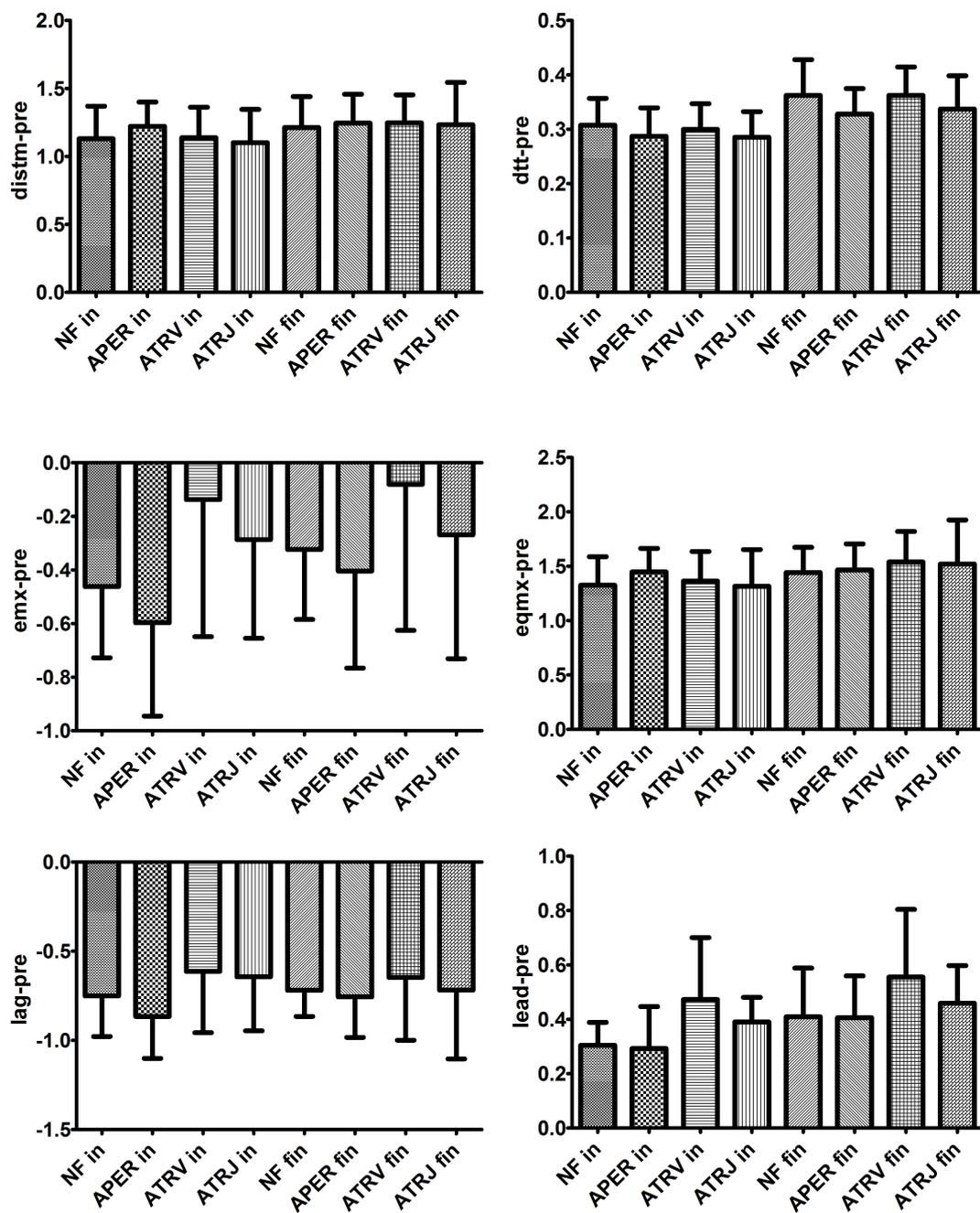


Figura 4.14: Grafici relativi all'analisi Pre-Delta-App del Pre_Test, con parametri considerati: distm, dtt, emx, eqmx, lag, lead. Tutte le grandezze sono state normalizzate rispetto al raggio del target tranne il dtt, normalizzato rispetto alla lunghezza del tratto percorso.

P-VALUE pre-deltap-app (t_test)				
	A-E	B-F	C-G	D-H
dism	0.1267	0.5718	0.0573	0.0141
dtm	0.0029	0.0010	0.0020	0.0010
emx	0.1881	0.0125	0.4973	0.8538
eqmx	0.1097	0.7404	0.0433	0.0040
lag	0.6710	0.0399	0.6291	0.3043
lead	0.0463	0.0028	0.0146	0.0908

P-VALUE pre-deltap-app (Wilcoxon)				
	A-E	B-F	C-G	D-H
dism	0.0674	0.8311	0.0537	0.0210
dtm	0.0006	< 0.0001	< 0.0001	0.0003
emx	0.3203	0.0186	0.3203	0.5186
eqmx	0.0840	0.9219	0.0488	0.0068
lag	0.7695	0.0645	0.2754	0.3652
lead	0.0020	0.0059	0.0195	0.1230

Figura 4.15: Tabella dei p-value relativa alle analisi 'Deltap-App' per il test Pre_Test

4.5.6 Pre-Tau

Dall'analisi del parametro Tau, definito alla sez. 4.1.10, è emerso che:

- l'*adaptation* alla variazione di scala è più rapida per il gruppo NF rispetto agli altri tre in termini di accuratezza (*dism*, *eqmx*). Il fatto che il tempo di adattamento aumenti in presenza del feedback potrebbe esser spiegato considerando che in questo caso il soggetto deve aggiornare anche il modello interno relativo al feedback, in quanto la variazione di scala determina un'alterazione del feedback (APER e ATRV) o una discrepanza tra feedback e movimento visualizzato (ATRJ).

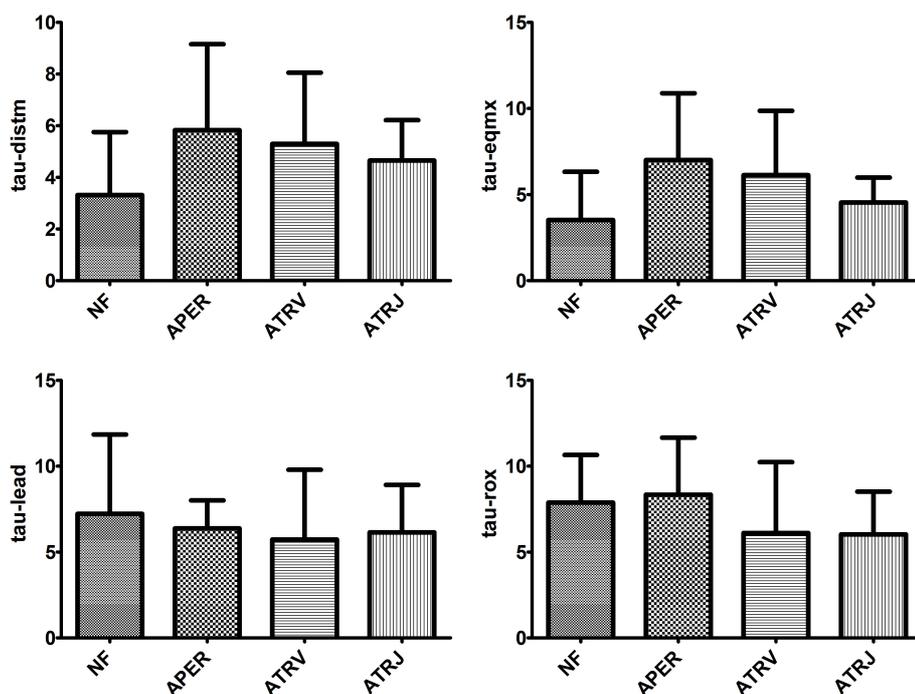


Figura 4.16: Grafici relativi all'analisi Pre-Tau del Pre_Test, con parametri considerati: *dism*, *eqmx*, *lead*, *rox*. Tutte le grandezze sono state normalizzate rispetto al raggio del target, tranne il *rox*.

P-VALUE pre-tau (Mann Whitney)						
	NF-APER	NF-ATRV	NF-ATRJ	APER-ATRV	APER-ATRJ	ATRV-ATRJ
distm	0.0702	0.0434	0.0185	0.9551	0.6691	0.8286
eqmx	0.0553	0.0434	0.0185	0.6943	0.2295	0.7618
lag	0.6623	0.4127	10000,0000	0.1143	0.6623	0.4127
lead	0.9314	0.6038	0.6607	0.2110	0.9682	0.4813
rox	0.8081	0.5035	0.4762	0.2766	0.1812	0.9546

P-VALUE pre-tau (Unpaired t-test)						
	NF-APER	NF-ATRV	NF-ATRJ	APER-ATRV	APER-ATRJ	ATRV-ATRJ
distm	0.0916	0.1248	0.1611	0.7421	0.3424	0.5389
eqmx	0.0480	0.1104	0.3223	0.6642	0.0842	0.2335
lag	0.4793	0.5806	0.7826	0.1381	0.5819	0.3081
lead	0.6095	0.4623	0.5388	0.6618	0.8296	0.7925
rox	0.8150	0.4553	0.3028	0.2412	0.1788	0.9679

P-VALUE pre-tau (Unpaired t-test with Welch's)						
	NF-APER	NF-ATRV	NF-ATRJ	APER-ATRV	APER-ATRJ	ATRV-ATRJ
distm	0.1195	0.1327	0.1645	0.7464	0.4141	0.5678
eqmx	0.0701	0.1280	0.3274	0.6657	0.1536	0.2897
lag	0.5077	0.5712	0.7850	0.1783	0.5839	0.3340
lead	0.6150	0.4661	0.5526	0.6514	0.8256	0.7931
rox	0.8051	0.3915	0.3239	0.2365	0.1640	0.9646

Figura 4.17: Tabella dei p-value relativa alle analisi 'Tau' per il test Pre_Test

4.6 Risultati Post_Test

4.6.1 Post-After

L'analisi denominata 'After' ha lo scopo principale di visualizzare l'after effect di una variazione avvenuta durante il task, e prende in considerazione il rapporto tra il comportamento tenuto dal soggetto nel ciclo di variazione della scala (cioè il 21° ciclo) rispetto all'andamento medio nei cicli precedenti (cioè il range che va dal 5° al 20° ciclo).

Le analisi concludono che:

- il ripristino della scala iniziale determina variazioni di performance e accuratezza comparabili nei quattro gruppi. Viene quindi meno l'apparente vantaggio derivante dall'uso di APER, riscontrato nel *Direct Effect*. Per interpretare questa differenza tra *Direct Effect* e *After Effect* possiamo considerare che tutti i soggetti hanno effettuato un esercizio maggiore, o che hanno già subito gli effetti di una prima variazione di scala: APER risulta utile soprattutto nelle situazioni inaspettate o nelle quali il soggetto ancora non ha ben appreso l'utilizzo dello strumento o il significato del feedback. Viceversa, i soggetti che hanno appreso l'uso dello strumento (NF) o che si sono abituati a ricevere il feedback relativo al task (ATRV e ATRJ), sono in grado di affrontare la seconda variazione con effetti simili a quelli di APER.

4.6.2 Post-Pin

L'analisi denominata 'Post-Pin' ha lo scopo principale di visualizzare l'andamento medio dei parametri nei primi cicli (dal 5° al 20°) del Post_Test e visualizzare il comportamento tenuto dal soggetto nella prima parte del secondo task sottoposto.

I risultati ottenuti con queste analisi fondamentalmente confermano quanto osservato nella parte dedicata al Pre-Pfin.

4.6.3 Post-PFin

L'analisi denominata 'Post-PFin' ha lo scopo principale di visualizzare l'andamento medio dei parametri nei cicli successivi al cambio scala (dal 25° al 39°) del Post_Test e visualizzare il comportamento tenuto dal soggetto in questo primo approccio con la nuova tipologia di scala proposta.

Dall'analisi si evince che il gruppo con audio legato all'errore di posizione (APER) sembra esser peggiorato sia in termini di performance(*emx*) che di accuratezza(*dism*), mentre il gruppo ATRV non presenta vantaggi rispetto agli altri gruppi.

Questo risultato entra in contrasto con l'analisi effettuata nella parte Pre-Pin, ma ciò può esser dovuto al fatto che è appena avvenuta una variazione rilevante.

Come proposta di lavoro futuro c'è la rielaborazione dei dati dal 30° ciclo, per consentire alla variazione di esaurirsi.

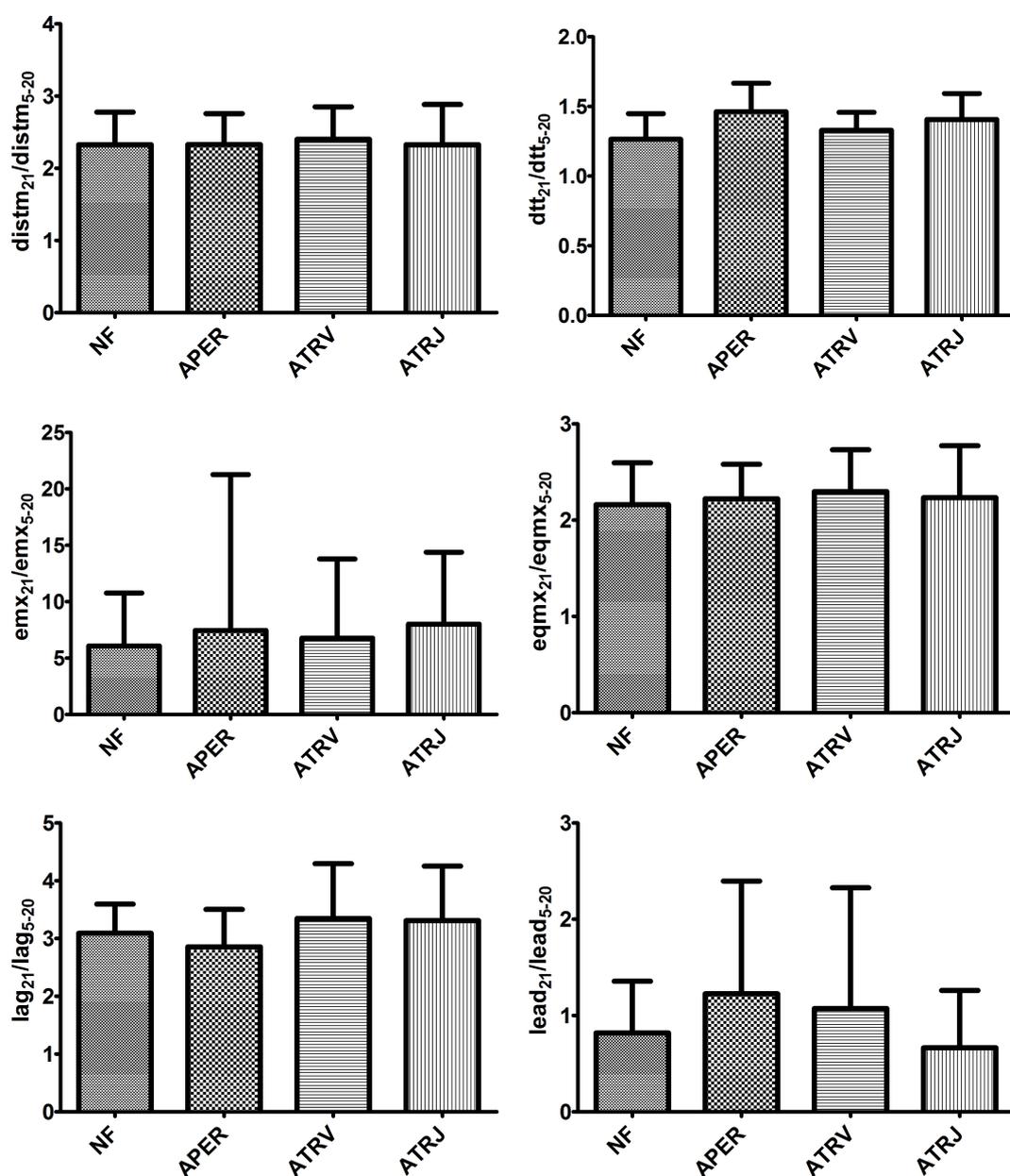


Figura 4.18: Grafici relativi all'analisi Post-After del Post_Test, con parametri considerati: distm, dtt, emx, eqmx, lag, lead. Tutte le grandezze sono state normalizzate rispetto al raggio del target tranne il dtt, normalizzato rispetto alla lunghezza del tratto percorso.

P-VALUE post-after (Mann Whitney)						
	NF-APER	NF-ATRV	NF-ATRJ	APER-ATRV	APER-ATRJ	ATRV-ATRJ
distm	0.7818	0.6009	0.9264	0.5444	0.9770	0.7075
dtm	0.0337	0.3099	0.0694	0.1939	0.8399	0.1939
emx	0.0289	0.9264	0.3099	0.3408	0.0531	0.3708
eqmx	0.9755	0.4417	0.7350	0.5067	0.9770	0.7508
lag	0.1661	0.7350	0.5181	0.1572	0.3123	0.9770
lead	0.5181	0.9755	0.3246	0.8852	0.3099	0.4061

P-VALUE post-after (Unpaired t-test)						
	NF-APER	NF-ATRV	NF-ATRJ	APER-ATRV	APER-ATRJ	ATRV-ATRJ
distm	0.9879	0.6940	0.9946	0.6904	0.9948	0.7251
dtm	0.0243	0.3365	0.0811	0.0696	0.4904	0.2510
emx	0.7576	0.7821	0.4191	0.8826	0.8985	0.6558
eqmx	0.7188	0.4632	0.7301	0.6496	0.9520	0.7536
lag	0.3428	0.4382	0.4993	0.1536	0.1809	0.9296
lead	0.3042	0.5418	0.5285	0.7610	0.1678	0.3369

P-VALUE post-after (Unpaired t-test with Welch's)						
	NF-APER	NF-ATRV	NF-ATRJ	APER-ATRV	APER-ATRJ	ATRV-ATRJ
distm	0.9880	0.6942	0.9945	0.6906	0.9948	0.7252
dtm	0.0241	0.3464	0.0817	0.0726	0.4907	0.2530
emx	0.7510	0.7787	0.4134	0.8831	0.8990	0.6560
eqmx	0.7214	0.4636	0.7277	0.6498	0.9520	0.7537
lag	0.3380	0.4288	0.4903	0.1558	0.1830	0.9297
lead	0.2951	0.5319	0.5289	0.7611	0.1617	0.3269

Figura 4.19: Tabella dei p-value relativa alle analisi 'After' per il test Post-Test

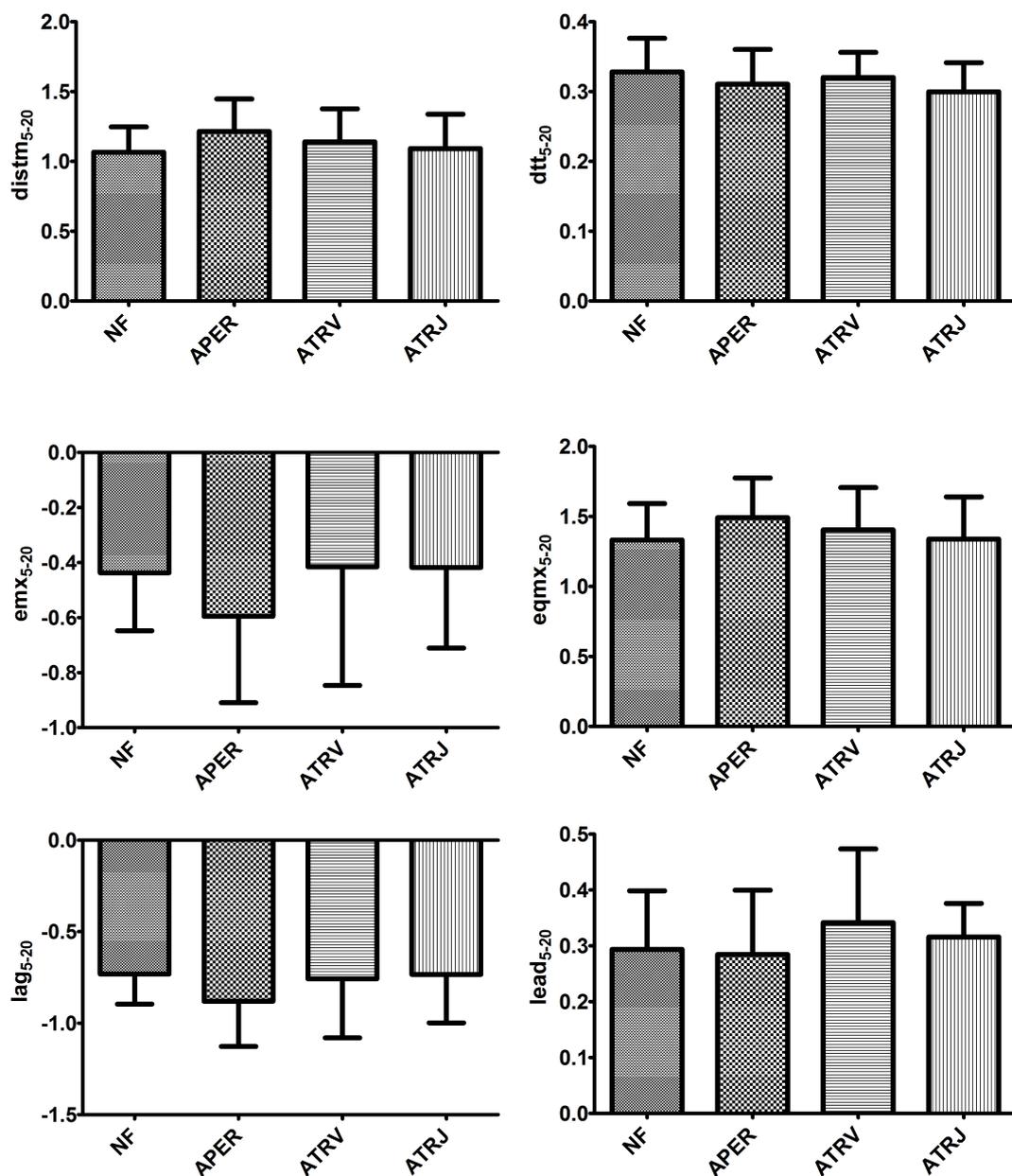


Figura 4.20: Grafici relativi all'analisi Post-Pin del Post_Test, con parametri considerati: $distrm$, dtt , emx , $eqmx$, lag , $lead$. Tutte le grandezze sono state normalizzate rispetto al raggio del target tranne il dtt , normalizzato rispetto alla lunghezza del tratto percorso.

P-VALUE post-pin (Mann Whitney)						
	NF-APER	NF-ATRV	NF-ATRJ	APER-ATRV	APER-ATRJ	ATRV-ATRJ
dism	0.1679	0.7427	0.9264	0.3579	0.2071	0.6444
dtm	0.6458	0.8955	0.0694	0.6936	0.5181	0.2071
emx	0.2122	0.5994	0.7350	0.2934	0.1481	0.9755
eqmx	0.1679	0.7928	0.8777	0.4701	0.1661	0.5181
lag	0.1891	0.8438	0.8777	0.2643	0.1661	0.9755
lead	0.8438	0.2643	0.5181	0.2372	0.3099	0.6009

P-VALUE post-pin (Unpaired t-test)						
	NF-APER	NF-ATRV	NF-ATRJ	APER-ATRV	APER-ATRJ	ATRV-ATRJ
dism	0.1098	0.4196	0.7753	0.4652	0.2356	0.6412
dtm	0.4178	0.6743	0.1465	0.6105	0.5684	0.2199
emx	0.1807	0.8834	0.8594	0.2771	0.1756	0.9884
eqmx	0.1869	0.5565	0.9626	0.4952	0.2254	0.6047
lag	0.1120	0.8123	0.9748	0.3291	0.1868	0.8508
lead	0.8453	0.3587	0.5357	0.2930	0.4144	0.5503

P-VALUE post-pin (Unpaired t-test with Welch's)						
	NF-APER	NF-ATRV	NF-ATRJ	APER-ATRV	APER-ATRJ	ATRV-ATRJ
dism	0.1115	0.4206	0.7725	0.4656	0.2351	0.6408
dtm	0.4183	0.6748	0.1506	0.6112	0.5721	0.2176
emx	0.1833	0.8840	0.8576	0.2785	0.1776	0.9886
eqmx	0.1876	0.5569	0.9623	0.4956	0.2248	0.6050
lag	0.1147	0.8134	0.9744	0.3304	0.1862	0.8523
lead	0.8455	0.3593	0.5476	0.2937	0.4309	0.5659

Figura 4.21: Tabella dei p-value relativa alle analisi 'PostIn' per il test Post_Test

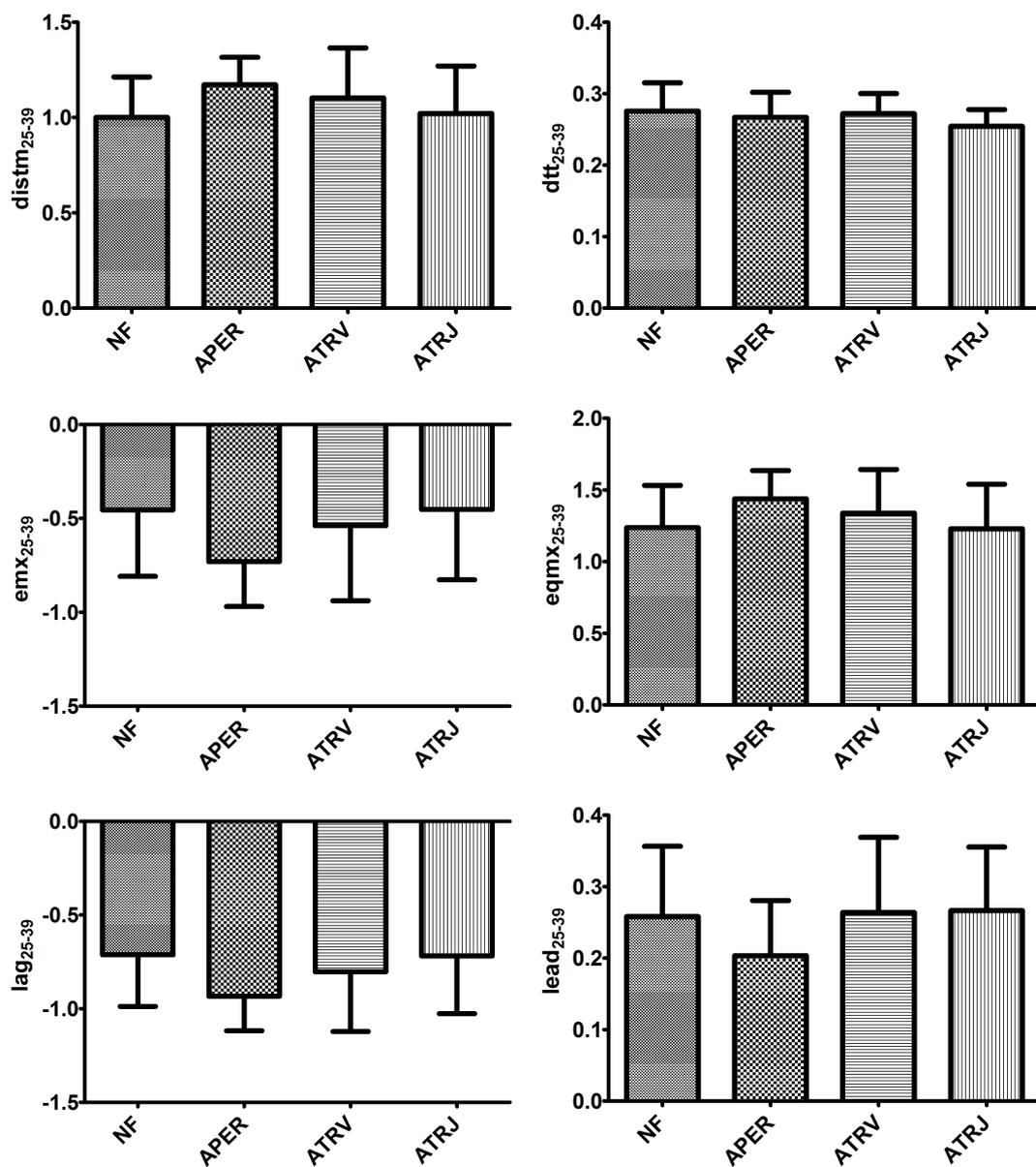


Figura 4.22: Grafici relativi all'analisi Post-PFin del Post_Test, con parametri considerati: distm, dtt, emx, eqmx, lag, lead. Tutte le grandezze sono state normalizzate rispetto al raggio del target tranne il dtt, normalizzato rispetto alla lunghezza del tratto percorso.

P-VALUE post-pfin (Mann Whitney)						
	NF-APER	NF-ATRV	NF-ATRJ	APER-ATRV	APER-ATRJ	ATRV-ATRJ
distm	0.0302	0.2934	0.9755	0.1679	0.0525	0.4061
dtm	0.7427		0.2301	0.6936	0.4417	0.1858
emx	0.0126	0.6458	0.9264	0.1486	0.0210	0.6444
eqmx	0.0356	0.2243	0.8777	0.1486	0.0392	0.2301
lag	0.0151	0.4307	0.9264	0.1150	0.0337	0.5588
lead	0.1150	0.7928	0.9755	0.1679	0.1166	0.9755

P-VALUE post-pfin (Unpaired t-test)						
	NF-APER	NF-ATRV	NF-ATRJ	APER-ATRV	APER-ATRJ	ATRV-ATRJ
distm	0.0386	0.3280	0.8353	0.4523	0.0939	0.4528
dtm	0.5869	0.8075	0.1252	0.7064	0.3169	0.1135
emx	0.0445	0.6051	0.9907	0.1876	0.0480	0.5980
eqmx	0.0745	0.4374	0.9506	0.3734	0.0702	0.4047
lag	0.0383	0.4852	0.9575	0.2514	0.0573	0.5281
lead	0.1629	0.8953	0.8322	0.1401	0.0854	0.9495

P-VALUE post-pfin (Unpaired t-test with Welch's)						
	NF-APER	NF-ATRV	NF-ATRJ	APER-ATRV	APER-ATRJ	ATRV-ATRJ
distm	0.0407	0.3287	0.8342	0.4552	0.0905	0.4543
dtm	0.5872	0.8078	0.1377	0.7066	0.3277	0.1179
emx	0.0467	0.6054	0.9907	0.1913	0.0462	0.5994
eqmx	0.0771	0.4379	0.9505	0.3753	0.0676	0.4048
lag	0.0404	0.4856	0.9573	0.2559	0.0547	0.5291
lead	0.1646	0.8953	0.8331	0.1418	0.0842	0.9500

Figura 4.23: Tabella dei p-value relativa alle analisi 'PostFin' per il test Post_Test

4.6.4 Post-Deltap

L'analisi denominata 'Post-Deltap' ha lo scopo principale di visualizzare la relazione tra le fasi del Post_Test che precedono e seguono la variazione della scala rispetto alla prima fase del test stesso; essa prende in considerazione il rapporto dato dalla differenza del comportamento tenuto dal soggetto nell'ultima fase del test (cioè dal 25° al 39° ciclo) con la prima fase (cioè dal 5° al 20° ciclo) rispetto a quest'ultima, cioè all'andamento medio nei primi cicli effettuati.

Qualunque sia la grandezza considerata, non si riscontrano differenze significative tra i diversi gruppi. Possiamo concludere che le differenze di performance nelle due condizioni di prova (scala iniziale e scala finale) non dipendono dal particolare tipo di feedback utilizzato.

4.6.5 Post-Deltap-App

L'analisi denominata 'Post-Deltap-App' ha lo scopo principale di visualizzare la relazione tra la fase iniziale del Post_Test (cioè dal 5° al 20° ciclo) e la fase finale della stessa fase (cioè dal 25° al 39° ciclo), cioè tra le due fasi che presentano una diversa scala video all'interno della stessa fase di test. Lo scopo è quello di visualizzare quale relazione intercorre tra i cicli precedenti e quelli che seguono la variazione della scala nei vari soggetti (è stata infatti utilizzata un'analisi per dati appaiati [sez.4.2.2]).

Valgono considerazioni analoghe quelle fatte nel Pre-Deltap-App, previa inversione dei segni delle variazioni.

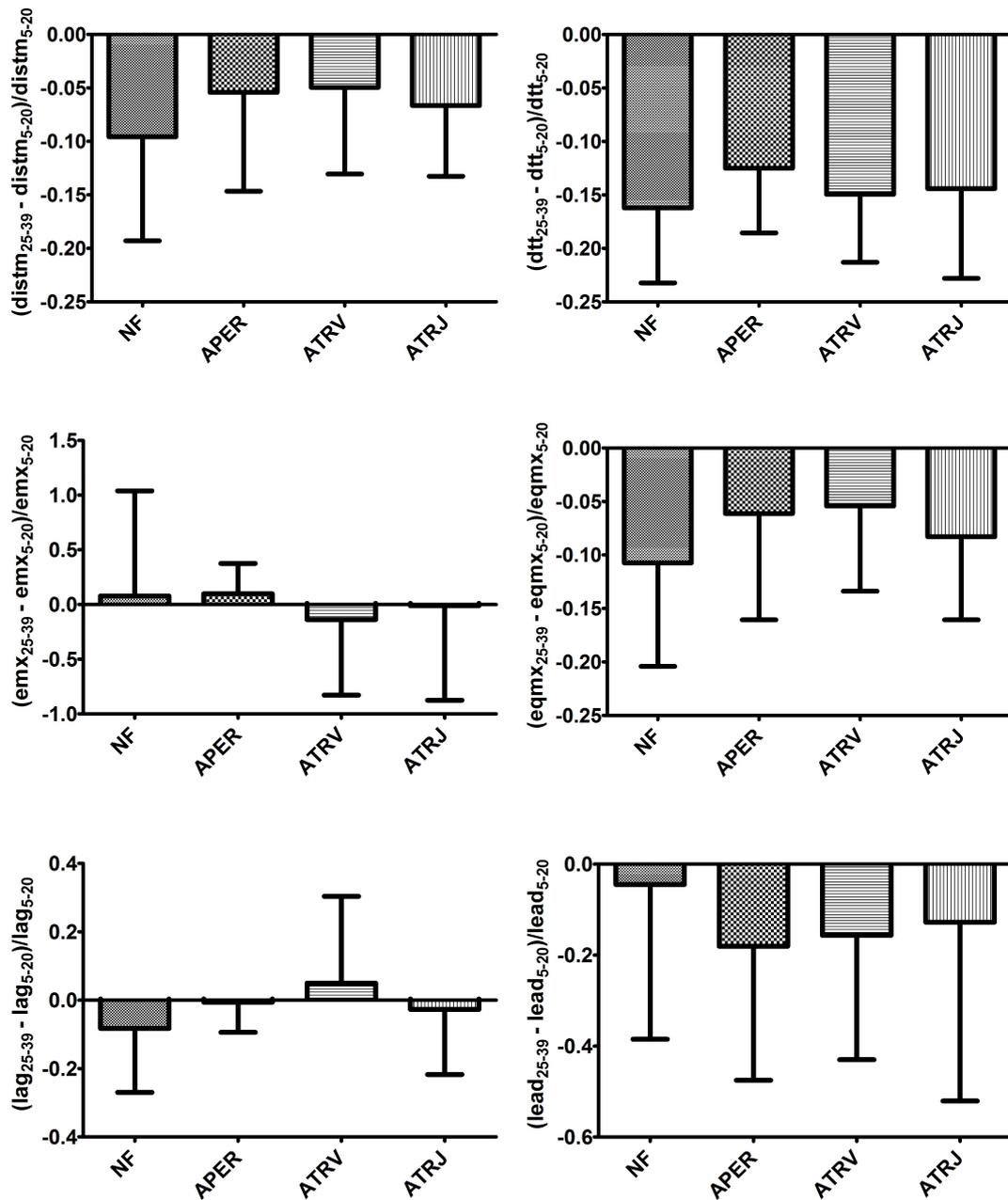


Figura 4.24: Grafici relativi all'analisi Post-Deltap del Post_Test, con parametri considerati: dism, dtt, emx, eqmx, lag, lead. Tutte le grandezze sono state normalizzate rispetto al raggio del target tranne il dtt, normalizzato rispetto alla lunghezza del tratto percorso.

P-VALUE post-deltap (Mann Whitney)						
	NF-APER	NF-ATRV	NF-ATRJ	APER-ATRV	APER-ATRJ	ATRV-ATRJ
distm	0.4002	0.3418	0.4483	0.6485	0.9717	0.7350
dtm	0.2110	0.5495	0.6209	0.3619	0.5458	0.8777
emx	0.3562	0.7513	0.4887	0.8197	0.5940	0.6444
eqmx	0.4470	0.2453	0.4887	0.8792	0.8590	0.3401
lag	0.1128	0.2453	0.4483	0.7040	0.9717	0.7350
lead	0.3154	0.6472	0.3390	0.7612	0.8590	0.9264

P-VALUE post-deltap (Unpaired t-test)						
	NF-APER	NF-ATRV	NF-ATRJ	APER-ATRV	APER-ATRJ	ATRV-ATRJ
distm	0.3556	0.2506	0.4138	0.9052	0.7255	0.5863
dtm	0.2357	0.6655	0.5932	0.3961	0.5720	0.8666
emx	0.9502	0.5619	0.8239	0.3508	0.7210	0.7040
eqmx	0.3198	0.1832	0.5174	0.8620	0.5815	0.3920
lag	0.2771	0.1932	0.4980	0.5404	0.7629	0.4194
lead	0.3680	0.4140	0.6058	0.8535	0.7399	0.8413

P-VALUE post-deltap (Unpaired t-test with Welch's)						
	NF-APER	NF-ATRV	NF-ATRJ	APER-ATRV	APER-ATRJ	ATRV-ATRJ
distm	0.3550	0.2564	0.4330	0.9067	0.7397	0.5903
dtm	0.2332	0.6673	0.5873	0.3943	0.5541	0.8650
emx	0.9483	0.5694	0.8260	0.3202	0.6888	0.7013
eqmx	0.3214	0.1891	0.5271	0.8655	0.5971	0.3930
lag	0.2656	0.1878	0.4977	0.5107	0.7403	0.4268
lead	0.3651	0.4201	0.6012	0.8549	0.7294	0.8390

Figura 4.25: Tabella dei p-value relativa alle analisi 'Deltap' per il test Post_Test

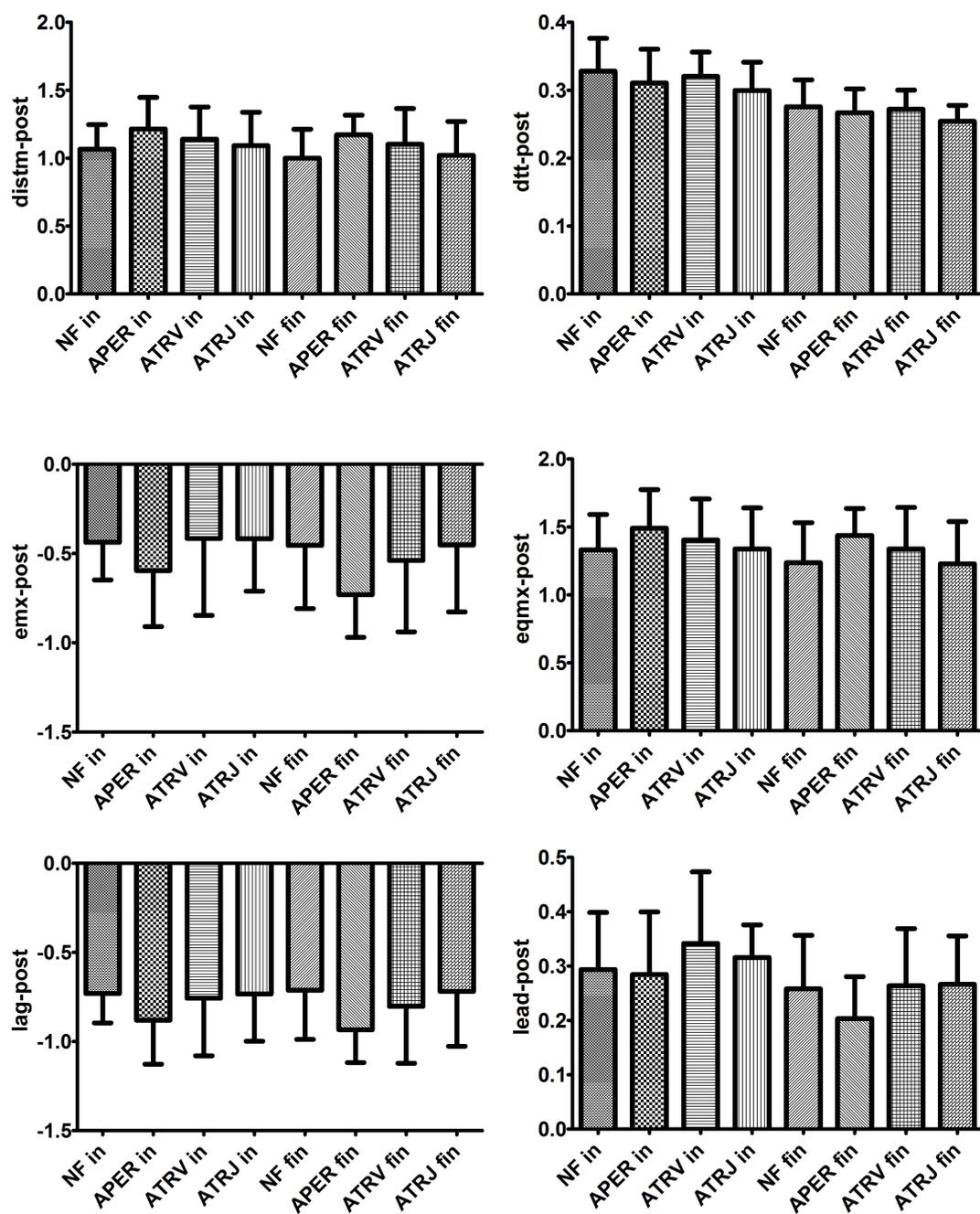


Figura 4.26: Grafici relativi all'analisi Post-Deltap-App del Post_Test, con parametri considerati: distrm, dtf, emx, eqmx, lag, lead. Tutte le grandezze sono state normalizzate rispetto al raggio del target tranne il dtf, normalizzato rispetto alla lunghezza del tratto percorso.

P-VALUE post-deltap-app (t_test)				
	A-E	B-F	C-G	D-H
dism	0.2767	0.3404	0.2455	0.0065
dtm	0.0010	0.0010	0.0010	0.0005
emx	0.8298	0.0428	0.1461	0.6071
eqmx	0.2296	0.3624	0.0997	0.0068
lag	0.7683	0.2256	0.3813	0.7139
lead	0.2090	0.0181	0.0369	0.1037

P-VALUE post-deltap-app (Wilcoxon)				
	A-E	B-F	C-G	D-H
dism	0.1475	0.3652	0.2061	0.0122
dtm	< 0.0001	0.0005	< 0.0001	0.0004
emx	0.9658	0.0830	0.1475	0.3013
eqmx	0.1230	0.3652	0.1230	0.0049
lag	0.3652	0.4131	0.3203	0.9697
lead	0.2402	0.0322	0.0537	0.0640

Figura 4.27: Tabella dei p-value relativa alle analisi 'Deltap-App' per il test Post_Test

4.6.6 Post-Tau

Dall'analisi del parametro Tau, definito alla sez. 4.1.10, è emerso che:

- come osservato in relazione all'*aftereffect*, la reazione alla seconda variazione di scala è omogenea nei quattro gruppi. Infatti, i tempi di adattamento in questo caso non sono diversi tra il gruppo NF e i gruppi che ricevono il feedback sonoro, a indicare come i soggetti abbiano appreso il significato del feedback. È anche ipotizzabile che la seconda variazione sia di natura diversa rispetto alla prima (infatti determina un ritardo invece che un anticipo - vedi *lead* e *lag* del Pre-Direct e del Post-After) e che questo determini un comportamento maggiormente omogeneo dei soggetti.

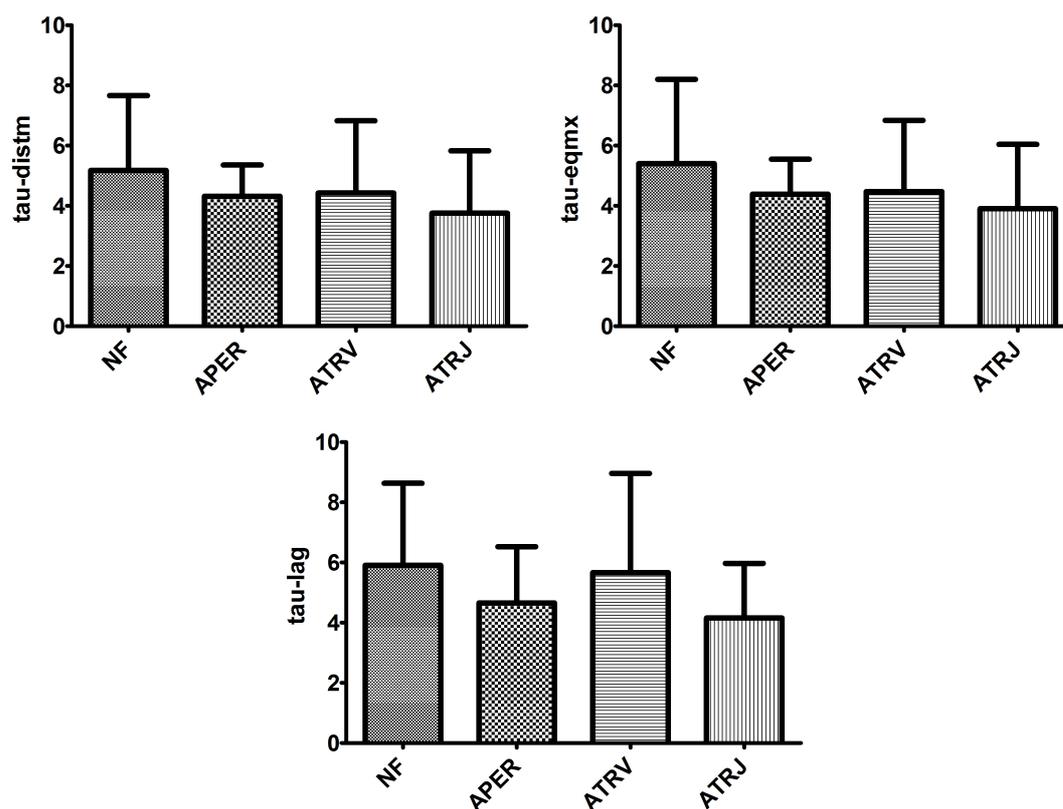


Figura 4.28: Grafici relativi all'analisi Post-Tau del Post_Test, con parametri considerati: distm, eqmx, lag. Tutte le grandezze sono state normalizzate rispetto al raggio del target.

P-VALUE post-tau (Mann Whitney)						
	NF-APER	NF-ATRV	NF-ATRJ	APER-ATRV	APER-ATRJ	ATRV-ATRJ
distm	0.3233	0.5545	0.2122	0.8197	0.7040	0.5994
eqmx	0.2875	0.5114	0.2372	0.9394	0.5433	0.5114
lag	0.2545	0.9476	0.1310	0.5949	0.7612	0.2934

P-VALUE post-tau (Unpaired t-test)						
	NF-APER	NF-ATRV	NF-ATRJ	APER-ATRV	APER-ATRJ	ATRV-ATRJ
distm	0.3491	0.4812	0.1624	0.9016	0.4713	0.4927
eqmx	0.3229	0.4102	0.1727	0.9203	0.5502	0.5575
lag	0.2593	0.8528	0.0915	0.4270	0.5501	0.1985

P-VALUE post-tau (Unpaired t-test with Welch's)						
	NF-APER	NF-ATRV	NF-ATRJ	APER-ATRV	APER-ATRJ	ATRV-ATRJ
distm	0.3195	0.4817	0.1633	0.8948	0.4456	0.4931
eqmx	0.2935	0.4107	0.1743	0.9153	0.5288	0.5579
lag	0.2427	0.8529	0.0942	0.4036	0.5517	0.2034

Figura 4.29: Tabella dei p-value relativa alle analisi 'Tau' per il test Post_Test

4.7 Confronto Pre/Post

4.7.1 Learning

L'analisi denominata 'Learning' ha lo scopo principale di visualizzare la relazione tra la fase iniziale del Pre_Test (cioè dal 5° al 20° ciclo) e la fase finale del Post_Test (cioè dal 25° al 39° ciclo), cioè tra le due fasi che presentano la medesima scala video 1:1. Lo scopo è quello di evidenziare somiglianze o meno nel comportamento del soggetto quando si ritrova ad effettuare il task nelle medesime condizioni, ma in momenti diversi all'interno del test.

Dai risultati è emerso che:

- in termini di accuratezza (*dt*) si nota un miglioramento tra fase iniziale e fase finale in tutti i gruppi, tranne APER(dato da verificare). Questo risultato indica che i soggetti hanno appreso l'utilizzo dello strumento.
- in termini di performance (*emx*) si nota un peggioramento per ATRV (dato da interpretare).

Sarebbe interessante ripetere le analisi per dati appaiati, dopo aver raggruppato tutti i soggetti in un unico gruppo.

4.7.2 Retention

L'analisi denominata 'Retention' ha lo scopo principale di visualizzare la relazione tra la fase finale del Pre_Test (cioè dal 35° al 49° ciclo) e la fase iniziale del Post_Test (cioè dal 5° al 20° ciclo), cioè tra le due fasi che presentano la medesima scala video 2:1. Lo scopo è quello di evidenziare somiglianze o meno nel comportamento del soggetto quando si ritrova ad effettuare il task nelle medesime condizioni, ma in momenti diversi all'interno del test.

- in termini di accuratezza (*dt*) si nota un miglioramento tra fase iniziale e fase finale in tutti i gruppi. Questo risultato indica che i soggetti hanno mantenuto la modifica del modello interno appreso a seguito della variazione di scala avvenuta nel Pre-Test, anzi hanno migliorato l'accuratezza.

- in termini di performance (*emx*) si nota un peggioramento per ATRV (analogamente al *Learning*) e per APER (dati da interpretare).

Sarebbe interessante ripetere le analisi per dati appaiati, dopo aver raggruppato tutti i soggetti in un unico gruppo.

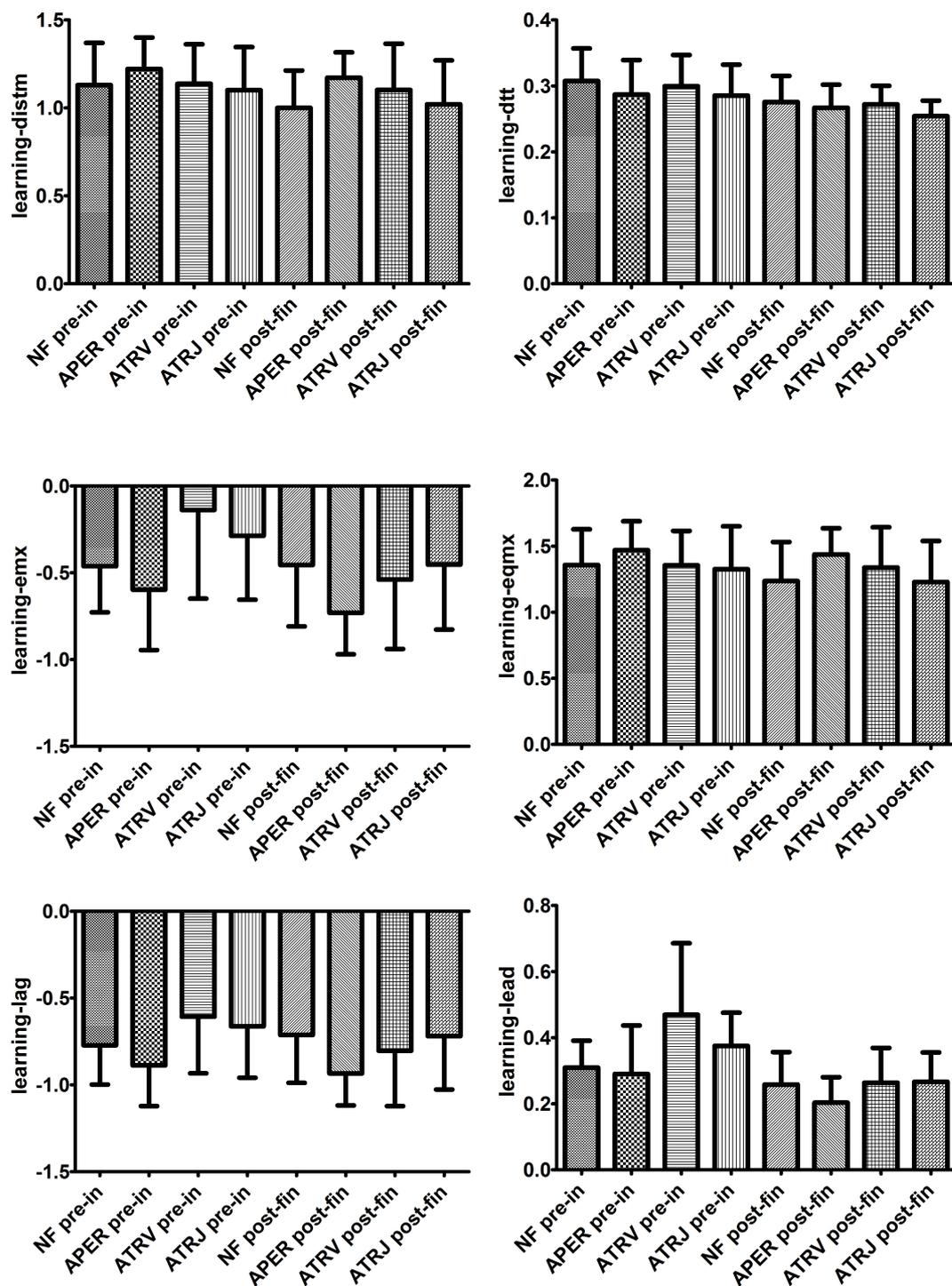


Figura 4.30: Grafici relativi all'analisi Learning nel confronto Pre-Post, con parametri considerati: distm, dtt, emx, eqmx, lag, lead. Tutte le grandezze sono state normalizzate rispetto al raggio del target tranne il dtt, normalizzato rispetto alla lunghezza del tratto percorso.

P-VALUE learning (t_test)				
	A-E	B-F	C-G	D-H
dstm	0.0779	0.2340	0.5154	0.2979
dtm	0.0029	0.1016	0.0098	0.0093
emx	0.9213	0.2042	0.0009	0.1791
eqmx	0.1610	0.5435	0.7968	0.3752
lag	0.3733	0.4584	0.0031	0.5503
lead	0.0661	0.0781	0.0027	0.0106

P-VALUE learning (Wilcoxon)				
	A-E	B-F	C-G	D-H
dstm	0.0537	0.0674	0.7002	0.2661
dtm	0.0017	0.1045	0.0068	0.0154
emx	0.7002	0.3652	0.0020	0.3394
eqmx	0.0830	0.1748	0.8311	0.3013
lag	0.3652	0.8311	0.0098	0.8501
lead	0.0674	0.1475	0.0020	0.0093

Figura 4.31: Tabella dei p-value relativa alle analisi 'Learning' per il confronto tra Pre_Test e Post_Test

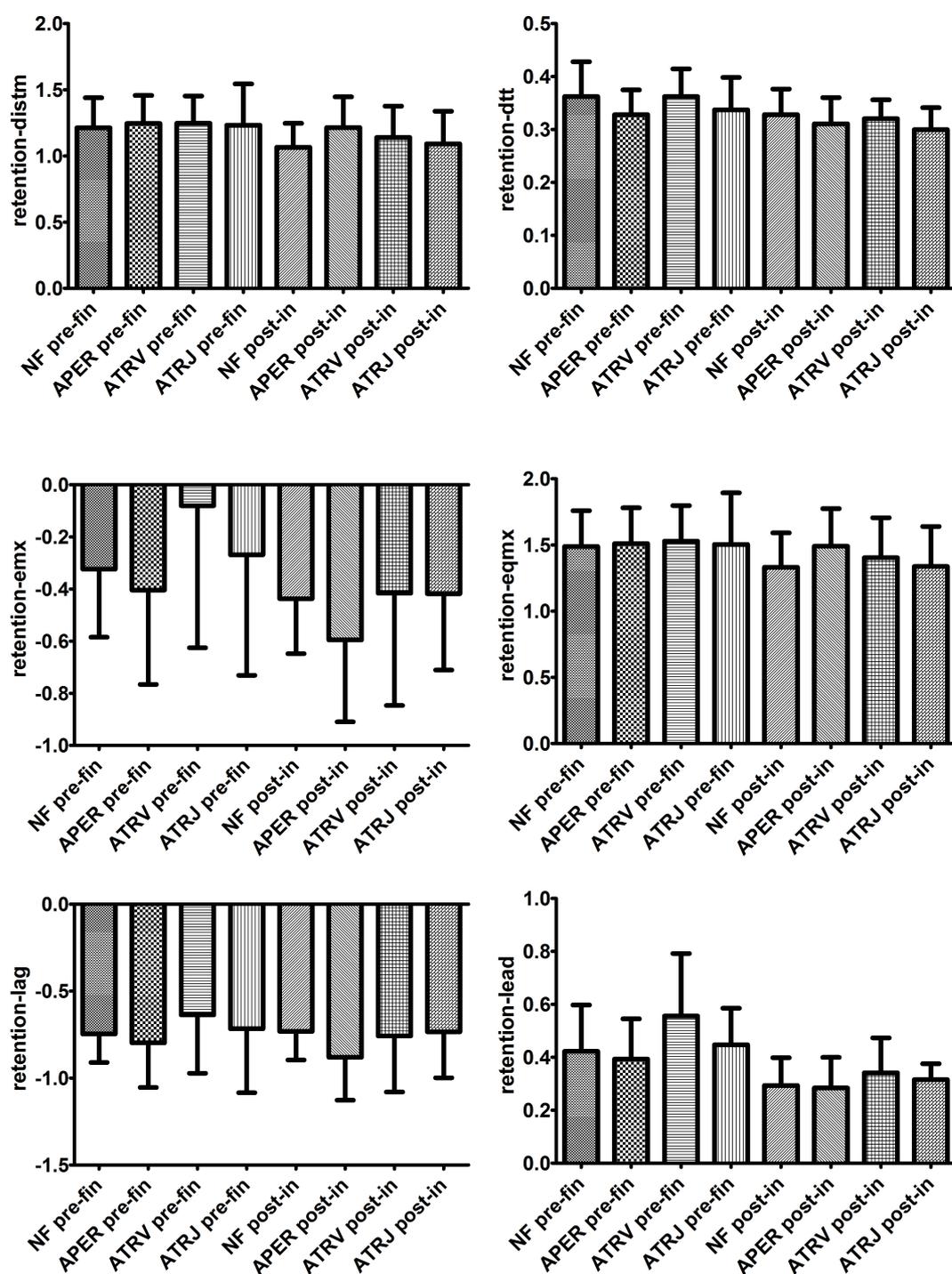


Figura 4.32: Grafici relativi all'analisi Retention nel confronto Pre_Post: distrm, dtt, emx, eqmx, lag, lead. Tutte le grandezze sono state normalizzate rispetto al raggio del target tranne il dtt, normalizzato rispetto alla lunghezza del tratto percorso.

P-VALUE retention (t_test)				
	A-E	B-F	C-G	D-H
distm	0.0215	0.4464	0.0924	0.0407
dtm	0.0537	0.0137	0.0010	0.0161
emx	0.2244	0.0508	0.0006	0.1169
eqmx	0.0398	0.7502	0.1167	0.0707
lag	0.7928	0.2216	0.0246	0.7935
lead	0.0210	0.0024	0.0006	0.0049

P-VALUE retention (Wilcoxon)				
	A-E	B-F	C-G	D-H
distm	0.0244	0.3652	0.0674	0.0522
dtm	0.0196	0.0159	0.0002	0.0118
emx	0.2783	0.0420	0.0049	0.1099
eqmx	0.0322	0.8311	0.0537	0.0640
lag	0.8984	0.2783	0.0322	
lead	0.0137	0.0098	0.0020	0.0068

Figura 4.33: Tabella dei p-value relativa alle analisi 'Retention' per il confronto tra Pre_Test e Post_Test

Conclusioni

Nell'ambito di questo lavoro sono stati effettuati dei test su soggetti sani per comparare differenti modalità di feedback sonoro, durante l'esecuzione di task motori.

Sono stati confrontati 4 gruppi di soggetti, uno senza feedback sonoro, uno con il feedback audio legato all'errore di posizione rispetto al target, uno con un audio legato alla velocità del target e l'ultimo con audio legato alla velocità del target scalata.

Durante i test la scalatura tra movimento effettuato dal soggetto e movimento visualizzato è stata variata, in modo da costringere il soggetto ad adattarsi a differenti condizioni di movimento.

È risultato che talune modalità di feedback possono aiutare il soggetto a migliorare la performance o l'accuratezza dei movimenti, anche se la quantità dei dati analizzati richiederà ulteriori analisi e verifiche per poter trarre delle conclusioni definitive.

Per i progetti futuri, sarebbe interessante 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] 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.
- [3] 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.
- [4] 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.
- [5] 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.
- [6] 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.

-
- [7] 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, pp. 323–327.
- [8] 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.
- [9] 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.
- [10] 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.
- [11] R. J. Nudo, "Postinfarct cortical plasticity and behavioral recovery," *Stroke*, vol. 38, no. 2, pp. 840–845, 2007.
- [12] 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.
- [13] World Health Organization, *The World health report 2008: Primary Health Care, Now More Than Ever*. WHO Press, Geneva, Switzerland, 2008.
- [14] 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.
- [15] T. N. Taylor *et al.*, "Lifetime cost of stroke in the united states," *Stroke*, vol. 27, pp. 1459–1466, 1996.

- [16] 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.
- [17] 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.
- [18] H. S. Jorgensen *et al.*, “Acute stroke with atrial fibrillation: The copenhagen stroke study,” *Stroke*, vol. 27, pp. 1765–1769, 1996.
- [19] J. H. Carr and R. B. Shepherd, *Stroke Rehabilitation - Guidelines for Exercise and Training to Optimize Motor Skill*. Butterworth-Heinemann, 2003.
- [20] Stroke Unit Trialists’ Collaboration, “Organised inpatient (stroke unit) care for stroke,” *Cochrane Database of Systematic Reviews*, no. 4, 2007.
- [21] P. Langhorne, F. Coupar, and A. Pollock, “Motor recovery after stroke: a systematic review,” *The Lancet Neurology*, vol. 8, pp. 741–754, 2009.
- [22] 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.
- [23] 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.
- [24] 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.

- [25] M. Butter *et al.*, *Robotics for Healthcare: Final Report*. European Commission - Directorate-General for Information Society and Media, October 2008.
- [26] 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.
- [27] 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.
- [28] J. Mehrholz, C. Werner, J. Kugler, and M. Pohl, “Electromechanical-assisted training for walking after stroke,” *Cochrane Database of Systematic Reviews*, no. 4, 2007.
- [29] 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.
- [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] 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.
- [32] 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.

-
- [33] 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.
- [34] 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.
- [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] 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.
- [37] Outpatient Service Trialists, “Therapy-based rehabilitation services for stroke patients at home,” *Cochrane Database of Systematic Reviews*, no. 1, 2003.
- [38] 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.
- [39] R. Wiles, A. Ashburn, S. Payne, and C. Murphy, “Discharge from physiotherapy following stroke: the management of disappointment,” *Social Science & Medicine*, vol. 59, no. 6, pp. 1263–1273, 2004.
- [40] C. Mah and F.A.Mussa-Ivaldi, “Evidence for a specific internal representation of motion-force relationships during object manipulation,” *Springer - Biological Cybernetics*, p. 13, 2003.
- [41] B.R.Rohrer, *Evolution of Movement Smoothness and Submovement Patterns in Persons with Stroke*. Department of Mechanical Engineering, Massachusetts Institute of Technology, 2002, ch. 2,3,7.

- [42] L.DiPietro, H.I.Krebs, S.E.Fasoli, B.T.Volpe, and N.Hogan, "Submovement changes characterize generalization of motor recovery after stroke," *Cortex*, vol. 3, p. 7, March 2009.
- [43] 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.
- [44] M.C.Cirstea and M.F.Levin, "Improvement of arm movement patterns and endpoint control depends on type of feedback during practice in stroke survivors," *Neurorehabil Neural Repair OnlineFirst*, p. 14, March 2007.
- [45] D. Feygin, M. Keehner, and R. Tendick, "Haptic guidance: experimental evaluation of a haptic training method for a perceptual motor skill," in *Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2002. HAPTICS 2002. Proceedings. 10th Symposium on*, 2002, pp. 40 – 47.
- [46] D. Morris, H. Tan, F. Barbagli, T. Chang, and K. Salisbury, "Haptic feedback enhances force skill learning," in *EuroHaptics Conference, 2007 and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. World Haptics 2007. Second Joint*, 2007, pp. 21 – 26.
- [47] 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.
- [48] G. Rosati, F. Oscari, D. J. Reinkensmeyer, R. Secoli, F. Avanzini, S. Spagnol, and S. Masiero, "Improving robotics for neurorehabilitation: enhancing engagement, performance, and learning with auditory feedback," in *Proceedings of the IEEE 12th International Conference on Rehabilitation Robotics ICORR2011*, Zurich, CH, June 29 - July 1 2011.
- [49] M. Battiston, "Studio dell'utilizzo del feedback uditivo nell'esecuzione di task motori," Master's thesis, Università degli studi di Padova, 2010.

Ringraziamenti

Il primo pensiero in questi brevi ringraziamenti va a mio padre e a mia madre, che mi hanno sempre seguito con grande passione in questo percorso universitario, sia a livello economico (che ovviamente è stato fondamentale), ma soprattutto mi son stati vicini e mi hanno sostenuto in tutti gli istanti, belli ma anche stressanti, di questo intenso periodo: non potrei essere più fortunata di così.

Un infinito grazie va alla mia sister Elisa, che più di ogni altra persona ha creduto in questa mia scelta di studi, sostenendomi in qualsiasi mia necessità o desiderio, anche economico, e che si è sempre entusiasmata per ogni mio piccolo passo verso questo traguardo, proprio come se ogni mia gioia e ogni mio successo fossero i suoi.

Ringrazio mia nonna, per il costante appoggio...e un grazie va anche a chi non c'è più e non potrà festeggiare con me questo bel momento, ma son sicura che da lassù gioiranno allo stesso modo per questo traguardo raggiunto. Ringrazio tutte le persone che hanno reso questi 3 anni così speciali..i miei cugini: Fabio, Giorgia, Mattia e anche i cugini 'acquisiti', ai quali son legata da un affetto che va oltre il semplice legame di parentela; le mie donne: Anna, Ilaria e Jessica, per tutti i momenti di chiacchiere, di confidenze, di consigli, di divertimenti e per la loro amicizia che è sempre presente e forte; le mie amiche 'Spice': Barbara, Erika, Samuela e tutte le altre, per le risate, le serate fantastiche (soprattutto nell'ultimo anno, davvero indimenticabile), la complicità e per un sacco di altre cose che loro fanno. Ringrazio i miei compagni di università, solo loro capiscono in pieno il percorso fatto finora e che ancora mi aspetta per i prossimi 2 anni; non faccio nomi, per evitare di dimenticare qualcuno, però un pensiero particolare va ad Alessandro, Anna e Matteo, con cui ho trascorso in modo davvero intenso questi ultimi 6 mesi e con cui, nonostante alcuni momenti di stanchezza, mi son divertita davvero tanto.

Un ringraziamento va inoltre al dottorando Fabio Oscari, che mi ha aiutato molto nel lavoro che ha portato alla stesura di questa tesi, non solo nel realizzare in modo pratico alcune parti dell'attività, ma soprattutto nell'aiutarmi a capire

concretamente i dettagli a me sconosciuti che si presentavano durante il lavoro; e un piccolo ringraziamento va anche al dottorando Andrea Carli, che ha reso possibile l'esecuzione dei numerosi test, mettendo a disposizione il suo ufficio e sopportando la nostra 'rumorosa' presenza.

Infine, il pensiero va al Prof. Giulio Rosati, nominato per ultimo in questi ringraziamenti, ma come lui ben sa...per me: '*Gli ultimi saranno i primi...*'.

Come promesso, non saranno ringraziamenti formali e di circostanza, ma davvero sentiti: un grazie per tutte le cose che mi ha insegnato, sia a livello di programmazione e di lavoro in gruppo, sia livello umano; un grazie per la immensa disponibilità, soprattutto durante gli intensi giorni dei test e nel periodo dei miei ultimi esami; un grazie per la costante e inesauribile attenzione alle mie necessità e alle mie richieste, di qualsiasi tipo e in qualsiasi momento queste si verificassero...e mille mille altri motivi per ringraziarlo, che lui conosce alla perfezione e non mi dilungo a scrivere qui...