

UNIVERSITÀ DEGLI STUDI DI PADOVA
FACOLTÀ DI INGEGNERIA



UNIVERSITÀ DEGLI STUDI DI PADOVA
FACOLTÀ DI INGEGNERIA

—
DIPARTIMENTO DI INNOVAZIONE MECCANICA E GESTIONALE
—

TESI DI LAUREA TRIENNALE IN INGEGNERIA BIOMEDICA

INFLUENZA DEL FEEDBACK
ESTRINSECO SULL'ESECUZIONE DI
TASK MOTORI

RELATORE: CH.MO PROF. ING. GIULIO ROSATI

LAUREANDO: ANNA LORENZIN

ANNO ACCADEMICO 2010-2011

Indice

Sommario	IX
Introduzione	XI
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	7
1.2 Motor Control e Motor Learning	10
1.3 Feedback	15
1.3.1 Visione e propriocezione	17
1.3.2 Feedback audio	18
1.3.3 Frequenza del feedback	20
1.4 Obiettivi	21
2 Sistema sviluppato	25
2.1 Architettura del sistema	25
2.2 Joystick a due gradi di libertà	26
2.2.1 Modello Simulink del joystick	28
2.3 Utilizzo dell'interfaccia GUI	32
2.4 Implementazione dell'interfaccia GUI	35
2.4.1 Creazione della traiettoria	35
2.4.2 Aggiornamento delle variabili (draw)	37
2.5 Feedback audio	41

2.5.1	Regole di creazione dei pacchetti - protocollo OSC	41
2.5.2	Pacchetti creati dall'interfaccia	43
2.5.3	Invio dei pacchetti	44
2.5.4	Software di generazione di suoni: Pure Data	45
2.6	Feedback visivo	46
3	Test Sperimentali	49
3.1	Test precedenti	49
3.2	Definizione del test	52
3.2.1	Feedback video	54
3.2.2	Feedback audio	55
3.3	Popolazione	58
3.3.1	Randomizzazione	59
3.4	Esecuzione dei test	59
3.4.1	Problemi riscontrati	60
4	Analisi statistica	63
4.1	Parametri calcolati per l'elaborazione	63
4.1.1	Distanza media dal target (dism)	64
4.1.2	Integrale del modulo della velocità relativa (dtt)	64
4.1.3	Ritardo medio (emx)	65
4.1.4	Errore medio di posizione in y (emy)	66
4.1.5	Errore quadratico medio in x (eqmx)	66
4.1.6	Errore quadratico medio in y (eqmy)	67
4.1.7	Lead error (lead)	67
4.1.8	Lag error (lag)	67
4.2	Test per l'analisi statistica	68
4.2.1	Wilcoxon test	68
4.2.2	t test	69
4.3	Elaborazioni dei dati	70
4.3.1	GraphPad Prism	71
4.3.2	Procedura	71
4.4	Risultati	73

4.4.1	Effetto del learning	77
	Conclusioni	83
	Bibliografia	85

Sommario

Lo scopo di questo studio è quello di investigare gli effetti che diversi tipi di feedback portano nell'esecuzione di alcuni esercizi di target tracking, confrontando in particolare la modalità video con quella audio. Per l'esecuzione dei test è stato utilizzato un joystick a due gradi di libertà, controllato da un software implementato in MatLab e Simulink. E' stata presa in considerazione una popolazione di 22 soggetti sani, privi di alcun tipo di menomazione motoria. Tre tipi di feedback audio e video sono stati sviluppati e forniti online durante l'esecuzione di sette esercizi di target tracking. Sono stati memorizzati dei dati durante i test per consentire la valutazione delle performance in momenti successivi; si sono, inoltre, effettuati dei test statistici per l'interpretare dei risultati ottenuti: sono stati comparati indici relativi ad esercizi di audio/video diverso ma eseguiti dallo stesso soggetto per la valutazione dell'effetto del feedback uditivo in paragone con quello video.

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 acutash/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, questi si differenziano in diversi modi, nel tipo di robot e di cinematica, nel livello di interazione robot-paziente (feedback), nel tipo di esercizio, nella durata della pratica e in base alle caratteristiche del paziente [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].

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].

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 in particolare la modalità video con quella audio. Per l'esecuzione dei test è stato utilizzato un joystick a due gradi di libertà, controllato da un software implementato in MatLab e Simulink.

Nel primo capitolo vengono presentati ed i risultati della ricerca bibliografica effettuata ed una revisione della letteratura, punto di partenza per l'ideazione e lo sviluppo di un nuovo tipo di test. Nel secondo capitolo viene introdotto il sistema sviluppato per l'implementazione dell'esercizio di tracking. Nel terzo capitolo viene descritta in dettaglio la procedura con cui sono stati effettuati i test. Nel quarto capitolo, infine, sono presentate le elaborazioni effettuate sui dati 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 certe informazioni: in particolare, verranno confrontati feedback audio e feedback video, i quali forniranno informazioni sulle stesse grandezze, seppur in 2 modalità differenti. Ci si aspetta che diverse modalità, visiva e sonora,

siano più adatte a rappresentare grandezze diverse. Inoltre verrà implementato un nuovo tipo di feedback, legato all'errore di velocità del soggetto rispetto al target, il quale si spera possa dare un contributo per una maggiore accuratezza nell'inseguimento del target.

Capitolo 1

Robotica in riabilitazione

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

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

Alla fine del capitolo vengono 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, questi si differenziano in diversi modi, nel tipo di robot e di cinematica, nel livello di interazione robot-paziente (feedback), nel tipo di esercizio, nella durata della pratica e in base alle caratteristiche del paziente [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 alla specifica robotica hardware utilizzata 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]. 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 si-

gnificativamente 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 movimenti stereotipati degli arti dei pazienti (tipicamente, traiettorie lineari in uno spazio di lavoro planare o 3D), spesso non legati alle attività funzionali incluse nelle misure ADL. Inoltre, viene tipicamente usata una realtà virtuale invece di una realtà mista e/o un contesto di training ambientale, questo può ostacolare il trasferimento dei risultati dei training alle situazioni di vita quotidiana [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.

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 attrattivo per le compagnie d'assicurazione e i sistemi di assistenza sanitaria, con l'obiettivo di creare un interesse commerciale a lungo termine nei sistemi di riabilitazione robotica. Inoltre, l'uso della tecnologia robotica sarà probabilmente estesa a differenti campi di riabilitazione, come la vasta gamma di malattie neuro-degenerative. Per esempio, sono stati presentati recen-

temente 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 significativamente 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 propriocezione, 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 movimen-

to, 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 al volo 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 (*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 terapista 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 perfor-

mance (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.

1.3.3 Frequenza del feedback

Un aspetto importante riguarda la frequenza con cui fornire tali feedback, poiché un sovraccarico di stimoli ostacola l'apprendimento. Fornire un alta frequenza può essere efficace nelle prime fasi di allenamento, tuttavia il soggetto corre il rischio di sviluppare una dipendenza dal feedback. La frequenza dipende innanzitutto dal livello motorio del soggetto; man mano che diventa più abile, i feedback saranno sempre più intervallati per permettere al soggetto di collegare le informazioni esterne a quelle interne sensoriali.

Mononen [48] ha condotto uno shooting test per studiare tale fenomeno. 58 maschi, con una limitata esperienza di tiro, hanno partecipato all'esperimento; diversi feedback sono state fornite con diverse frequenze:

- VIS100: feedback visivo, dopo ogni prova
- VIS50: feedback visivo, dopo il 50% delle prove
- AUD50: feedback audio, dopo il 50% delle prove
- KR: dopo ogni prova
- CO: gruppo di controllo, senza feedback

La bontà della performance è stata valutata in termini di precisione del tiro e di stabilità dell'arma. I soggetti hanno condotto un periodo di allenamento di 4 settimane, durante il quale sono stati forniti i feedback; la prova è stata ripetuta, senza l'ausilio di feedback, 2, 10 e 40 giorni dopo per verificare l'apprendimento (*retention test*). Nonostante i risultati abbiano indicato una maggiore efficacia del VIS100 sul VIS50 durante la fase di allenamento, tuttavia, tale superiorità è

scomparsa progressivamente durante il retention test. Invece, il gruppo a cui è stato fornito VIS50 è stato in grado di mantenere gli stessi punteggi fino alla fine del test; ciò suggerisce che i soggetti appartenenti a tale gruppo sono stati capaci di costruire uno schema motorio duraturo, che limita la dipendenza dal feedback e che permette l'esecuzione dell'esercizio anche senza tale ausilio. Comunque, i punteggi migliori sono stati ottenuti dal gruppo a cui è stato fornito AUD50, sia durante l'allenamento, sia durante il retention test. Tale stimolo ha, infatti, creato un ovvio collegamento tra il feedback fornito e la correzione del movimento che il soggetto deve operare; tale informazione non è, invece, di così facile acquisizione con il solo ausilio del feedback visivo.

1.4 Obiettivi

Lo scopo di questo lavoro è quello di investigare gli effetti che diversi tipi di feedback, audio e video, 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, come può essere l'errore di velocità o di posizione del paziente rispetto al target. Grande attenzione è stata rivolta al ruolo del feedback audio, 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 22 soggetti sani, privi di alcun tipo di menomazione motoria. Diversi tipi di feedback audio e video 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 e di velocità del soggetto rispetto al target e alla velocità del target. Tali feedback sono stati somministrati sia in modalità visiva, attraverso il cambiamento del colore dello sfondo del monitor, sia in modalità sonora, attraverso l'ascolto di particolari suoni.

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.

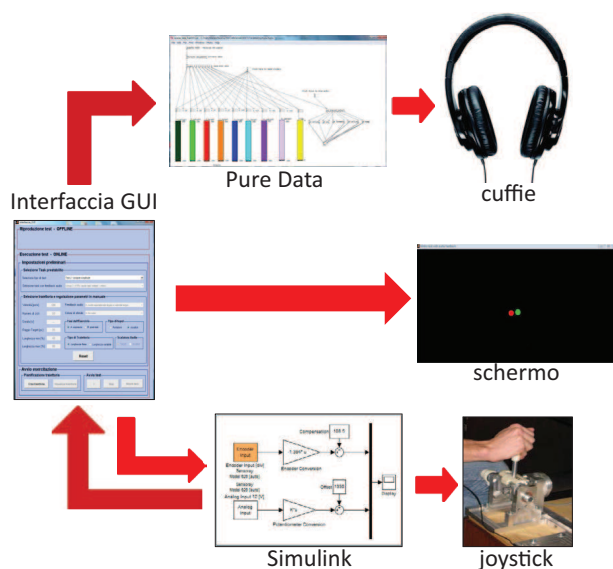


Figura 2.2: Schema che riassume i collegamenti logici tra le varie parti del sistema

2.2 Joystick a due gradi di libertà

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

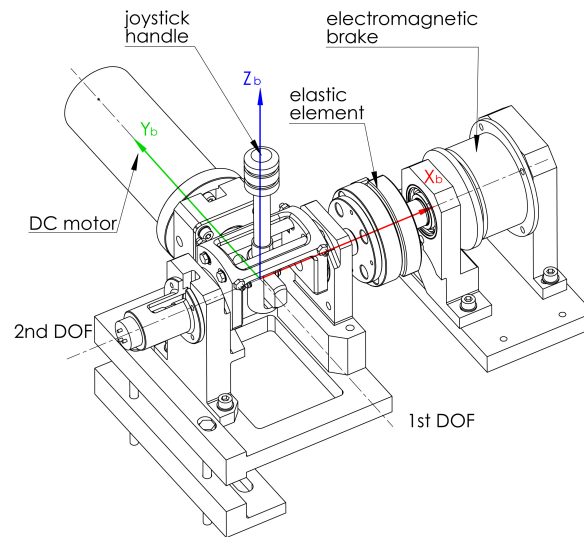


Figura 2.3: Immagine CAD del joystick

corrente continua, e un g.d.l. passivo, comandato da un freno elettromagnetico onoff 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 finecorsa, mentre il movimento di rollio dai bordi della cava realizzata sul supporto; il range di movimento per entrambi gli assi è limitato a circa $\pm 30^\circ$. Per agevolare i movimenti dei soggetti è stata realizzata una prolunga in plastica della leva di 450 mm, così da ottenere una lunghezza totale della leva pari a $l = 160 \text{ mm}$; l'arco, cioè lo spazio percorso dall'estremità della leva è pari a: $\pi/3 l = 167,55 \text{ mm}$, la corda da esso sottesa risulta: $2 \sin(\pi/6) l = 160 \text{ mm}$.

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 $10\text{k}\Omega$ e alimentato a 10 V . La conversione dei dati avviene mediante un ADC a 16 bit, impostato su un range di $\pm 10\text{V}$ (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.

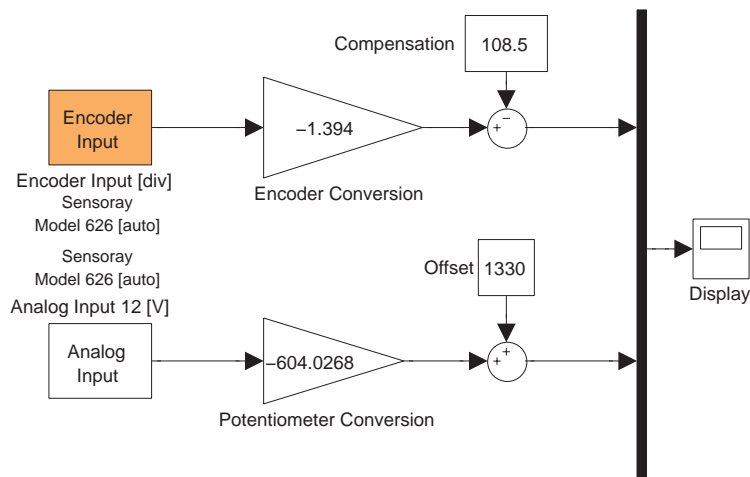


Figura 2.5: Schema del modello Simulink del joystick

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.

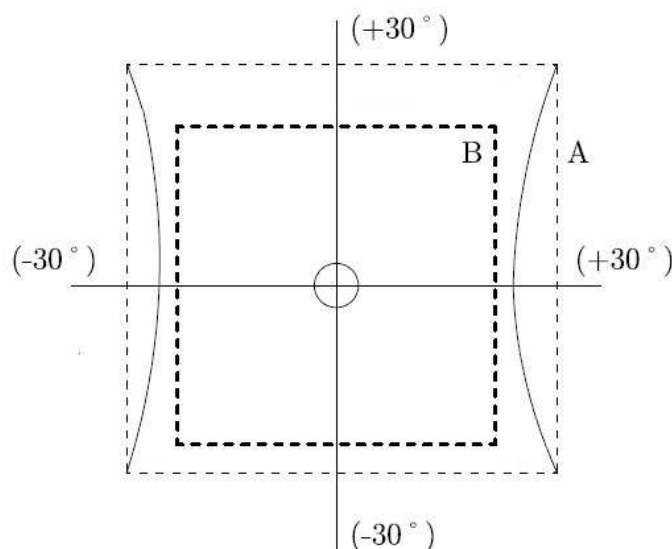


Figura 2.6: Area di lavoro della leva del joystick con evidenziati i limiti individuati tramite la procedura di calibrazione (A = limiti massimi dell'area di lavoro, B = area di lavoro effettiva)

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 *eventListener*, ed è direttamente implementata nell’interfaccia GUI.

Le istruzioni relative alla memorizzazione dei dati sono riportate nelle seguenti righe di codice:

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

Il tempo della simulazione è generato dall’istruzione:

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

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

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

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

```
function mdl = loadModel(modelName)
% 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

Per agevolare tutte le operazioni durante l'esecuzione dei test sui soggetti è stata realizzata una GUI (*graphical user interface*) in MatLab. Essa 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.

L'interfaccia è stata sviluppata in modo da poter gestire 2 tipi di test: la selezione è effettuata utilizzando la prima tendina di selezione 'test menù' è at-

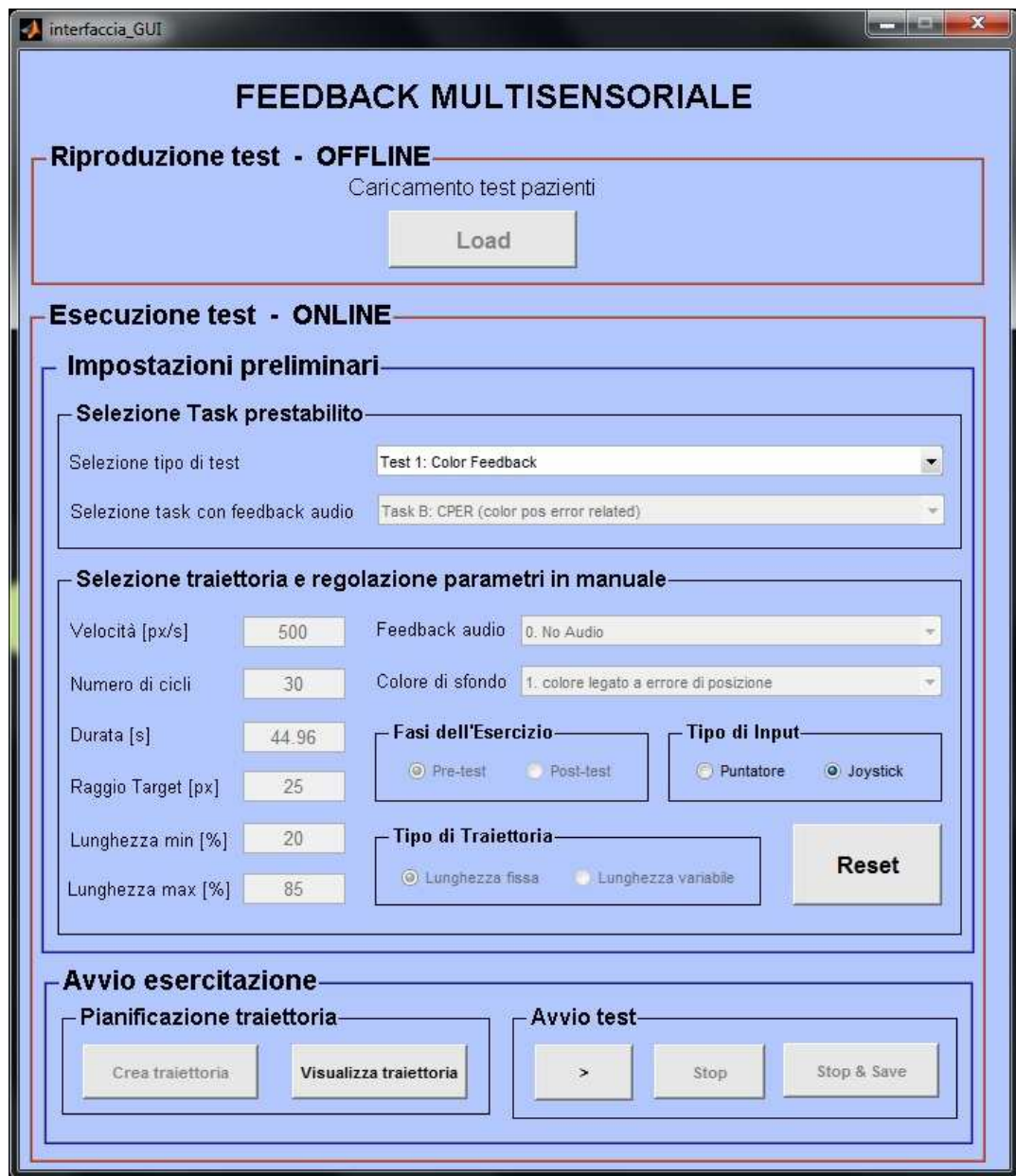


Figura 2.7: Interfaccia GUI implementata

traverso la quale è possibile scegliere tra ‘*Color Feedback*’ e ‘*Variable Amplitude*’. In questo contesto verrà descritto solamente il primo.

La sezione dell’interfaccia destinata alla creazione dell’esercizio prevede degli appositi controlli per stabilire nei minimi dettagli tutte le caratteristiche del test che si sta creando. Si può procedere in 2 modi differenti: creare l’esercizio manualmente o caricare task preimpostati.

Per quanto riguarda la creazione del task manualmente, dopo aver selezionato dal primo menù a tendina ‘*Test 1: Color Feedback*’, si può procedere direttamente nella scelta di tutti i parametri desiderati:

- Velocità [px/s]: consente di scegliere un qualsiasi valore di velocità con cui si vuole fare muovere il target
- Numero di cicli: imposta il numero di cicli da fare percorrere al target
- Durata [s]: questo campo non prende un input esterno ma, una volta impostato tutto il setting, permette di visualizzare la durata del task
- Raggio target [px]: stabilisce la dimensione del target
- Lunghezza min [%]: imposta, in percentuale rispetto alla grandezza della finestra, l’ampiezza minima di movimento che il target può compiere durante l’esercizio
- Lunghezza max [%]: imposta, in percentuale rispetto alla grandezza della finestra, l’ampiezza massima di movimento che il target può compiere durante l’esercizio
- Feedback audio: menù a tendina che permette la selezione di un feedback audio da associare all’esercizio
- Colore di sfondo: menù a tendina che permette la selezione di un eventuale feedback visivo che consiste nella variazione del colore di sfondo dello schermo
- Tipo di traiettoria: finestra di selezione che stabilisce se il target si muove di un tratto variabile o costante (‘*Test 1: Color Feedback*’ la selezione della lunghezza minima non è abilitata)
- Tipo di input: permette di scegliere lo strumento da utilizzare per l’esecuzione del task (se puntatore o joystick)
- Fasi dell’esercizio: per ‘*Test 1: Color Feedback*’ questo comando non è abilitato.

Dopo aver selezionato tutti i parametri, è necessario premere il bottone ‘*Crea traiettoria*’; contemporaneamente vengono disabilitati tutti i campi di selezione compresi nel riquadro di selezione e regolazione (tranne quello del puntatore) e vengono abilitati i pulsanti ‘*Visualizza traiettoria*’ e ‘>’ (play): il primo consente

di visualizzare in una finestra il grafico della traiettoria appena creata, il tasto *'Play'* avvia l'esercizio.

Non appena parte lo svolgimento del task, nel riquadro in basso si abilitano i tasti *'Stop'* e *'Stop & Save'*: entrambi servono per stoppare l'esercizio in qualsiasi istante, il secondo, in più, consente di salvare i dati acquisiti durante lo svolgimento del task.

Una volta terminato l'esercizio, per procedere con un'altra selezione, bisogna prima premere il tasto *'Reset'* che ristabilisce i parametri di default e riabilita tutti i campi.

Per agevolare la procedura di lavoro, è stata utilizzata la tendina *'Task menù'* che permette il caricamento di esercizi prestabiliti, di cui sono già state scelte tutte le impostazioni. Una volta scelto il task desiderato, i campi vengono automaticamente disabilitati e riempiti con i valori preimpostati; si può, quindi, procedere direttamente all'avvio dell'esercizio senza dover fare alcuna scelta. Così facendo, si riducono i tempi di attesa tra un task ed il successivo e si limita il rischio di eventuali errori di selezione da parte dell'operatore.

2.4 Implementazione dell'interfaccia GUI

L'interfaccia GUI gestisce le varie fasi della creazione ed esecuzione del task permettendo la comunicazione tra Matlab e 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 *evtListener* che permette di eseguire le istruzioni contenute nella *draw* in maniera ripetitiva e sincronizzata.

2.4.1 Creazione della traiettoria

La funzione *build_trajectory2* crea gli estremi della traiettoria: questa è rappresentata da un vettore che contiene 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 po-

sizione 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 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 che specifica istanti di tempo iniziali e finali, oltre a posizioni, velocità ed accelerazioni. Il tratti presentano un simmetrico profilo di velocità a campana con una caratteristica globale di dolcezza della traiettoria che è ben approssimata dal criterio di 'minimum-jerk'.

La funzione *crea_tra*, accessibile direttamente all'interno della GUI, sfrutta le funzioni *build_trajectory2* e *traj_pp* per costruire la traiettoria desiderata. In particolare, salva la struttura 'exercise_parameters' che contiene tutti i parametri che identificano il task.

Il primo campo è la matrice 'exercise_parameters.data', nella quale vengono salvate tutte le variabili relative al soggetto e al target, le quali verranno aggiornate ogni volta che l'*EventListener* richiama la funzione *draw* durante l'esecuzione dell'esercizio. Inoltre vengono salvati:

- velocità del target
- numero di cicli
- lunghezza minima
- lunghezza massima
- raggio del target
- dimensioni dello schermo
- tipo di traiettoria selezionata
- stringa che identifica la traiettoria selezionata

- grado della funzione interpolatrice
- etichette che identificano le variabili di *exercise_parameters.data*
- tipo di test
- tipo di task
- istanti di inizio ciclo.

2.4.2 Aggiornamento delle variabili (draw)

La funzione *draw* disegna soggetto e target in ogni istante temporale, salvando i dati aggiornati relativi all'esercizio nella matrice *exercise_parameters*; permette inoltre la comunicazione con il software dei suoni attraverso il meccanismo della spedizione dei pacchetti ed imposta, se necessario, il colore dello schermo. Viene lanciata, alla frequenza di 100Hz, dalla funzione *eventListener*, implementata direttamente nell'interfaccia GUI, attraverso l'istruzione:

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

e riceve in ingresso:

- 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)

La GUI passa come parametro temporale t_att: questo 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 (pre-esercizio), per cui all'avvio dell'esercizio t_att parte da -10s. Un secondo vettore, t_sim, definisce i tempi di simulazione nei quali il target effettivamente si muove.

L'indice `indt` permette di scorrere `t_sim`: esso viene posto a 1 nella fase di pre-esercizio e da quando `t_att` è pari a zero inizia a scandire `t_sim` con un periodo di campionamento pari a 0,01s, fino a quando il target non si ferma (esercizio). Al termine dell'esercizio il vettore `t_sim` non viene più incrementato e `indt` assume un valore pari a `end`, ultima posizione del vettore (post-esercizio).

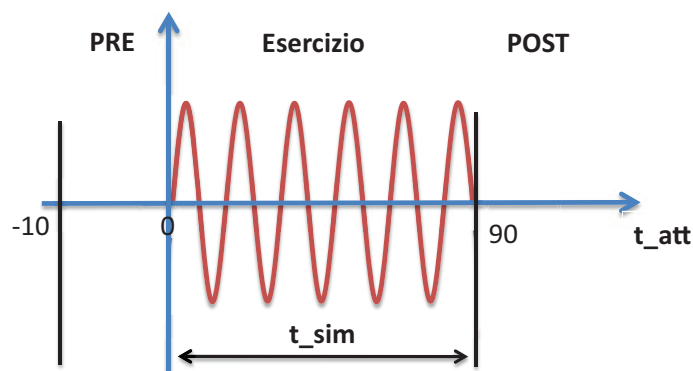


Figura 2.8: Relazioni tra i vari tempi utilizzati all'interno della `draw`

Il primo passo consiste nel disegnare l'indicatore del soggetto (pallino verde), sfruttando le coordinate di posizione presenti nella struttura `sim_data` presente in ingresso. La `draw` prevede anche la possibilità di applicare un'eventuale scalatura video alle coordinate delle posizioni; queste vengono moltiplicate per un fattore che ne riduce o amplifica la scala. In questo test non è stata utilizzata tale funzione, per cui il fattore di scalatura è stato impostato a 1.

La fase di pre-esercizio permette al soggetto di posizionarsi al centro dello schermo prima che il target inizi a muoversi. All'avvio dell'esercizio precede un countdown costituito da una successione di 'beep' della durata di 0.5 secondi intervallati da altrettante pause della stessa durata. Questa operazione è realizzata attraverso l'invio al PureData di pacchetti con tutti i campi a zero tranne quello relativo all'audio, il quale è composto da una successione correttamente temporizzata di 0 e 9.

Nella fase di esercizio viene innanzitutto disegnato l'indicatore del target (pallino rosso). Poiché la velocità del soggetto non è disponibile a partire dalla GUI,

viene calcolata attraverso la derivata numerica: in particolare, lo spazio percorso dal soggetto in un intervallo di tempo pari a 10 volte il periodo di campionamento (0,01s) viene diviso per l'intervallo di tempo stesso. Successivamente, solo quando il task lo prevede, viene impostato il colore dello schermo; a seconda della modalità di feedback video prevista, viene calcolato il parametro corrispondente:

- errore di posizione, pari alla distanza euclidea normalizzata rispetto a 6 volte il raggio del target
- errore di velocità, pari alla differenza di velocità tra soggetto e target, normalizzata rispetto a 1,5 volte la velocità massima del target
- velocità del target, normalizzata rispetto a 2 volte la velocità massima del target

Il colore dello sfondo è definito attraverso il modello RGB (red, green, blue) che imposta lo sfondo miscelando tra loro i tre colori a disposizione. Il colore dello sfondo è definito da un vettore di dimensioni 1x3, dove le tre celle rappresentano rispettivamente l'intensità del rosso [1 0 0], del verde [0 1 0] e del blu [0 0 1] e possono variare in un intervallo compreso tra zero e uno; il colore nero corrisponde alla combinazione [0 0 0]. Si è scelto di far variare il colore di sfondo tra il colore nero (colore di base) e l'azzurro [0 0.5 0.5] (fondo scala); è stata inoltre impostata una soglia, in questo caso posta pari a zero, sotto la quale lo sfondo rimane nero. In base al valore del parametro calcolato in precedenza, il colore di sfondo viene di volta in volta impostato nel seguente modo:

- se $err < soglia$: `set(gca, 'Color', [0 0 0])`
- se $err \leq 1$: `set(gca, 'Color', errN*colore)`
- altrimenti: `set(gca, 'Color', colore)`

dove err è il parametro calcolato in precedenza, mentre $errN$ risulta da modifica applicata per migliorare l'effetto di gradazione del colore e vale:

$$errN = \frac{(err - soglia)^3}{(1 - soglia)} \quad (2.3)$$

$errN$ è un numero compreso tra zero e uno: esso viene moltiplicato per l'RGB del fondo scala, per cui quando $errN$ è nullo lo sfondo rimane nero, altrimenti questo

sfuma in maniera proporzionale al valore di $errN$, fino ad un massimo di $[0 \ 0.5 \ 0.5]$.

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.

Nel post-esercizio, in modo analogo a quanto accade nel pre, viene implementato un unico beep che segnala il termine dell'esercizio; lo sfondo viene resettato a nero e le velocità poste a zero.

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) xt看V 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

La variabile *counter* corrisponde al contatore dei dati salvati ed è pari al numero di cicli del modello Simulink.

2.5 Feedback audio

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

2.5.1 Regole di creazione dei pacchetti - protocollo OSC

L'Open Sound Control (OSC) è un protocollo di trasmissione che permette a strumenti musicali, computer ed altri dispositivi multimedia di scambiare 'music performance data' in tempo reale attraverso una semplice rete interna o Internet. L'OSC rappresenta un'evoluzione del MIDI (Musical Instrument Digital Interface), ovvero il protocollo standard, utilizzato fin dagli anni 80, per l'interazione degli strumenti musicali elettronici. La pressante domanda di riduzione dei costi,

maggiore affidabilità e maggiore accessibilità a tutti gli utenti ha portato l'OSC a svilupparsi a tal punto da eguagliare e poi superare l'uso del MIDI.

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

Tipi di dati fondamentali del protocollo OSC

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

OSC packets

Gli OSC packets, più comunemente noti come pacchetti, sono l'unità fondamentale di trasmissione del protocollo OSC. Tali pacchetti sono costituiti sia dai dati in essi contenuti scritti in formato binario, sia dalla dimensione di questi dati, tale dimensione deve essere sempre un numero di byte multiplo di 4. Inoltre, nell'atto di trasportare i pacchetti la rete deve tenere conto sia delle informazioni relative al contenuto sia di quelle relative alla dimensione, che devono essere entrambe

trasmesse. I dati che vengono trasmessi in un pacchetto, riferito al protocollo OSC, si dividono in 2 tipologie: messaggio (message) o fascio (bundle). Il primo byte di dati permette di definire se il contenuto del pacchetto è del primo o del secondo tipo.

OSC Messages: è costituito da un OSC Address Pattern, ovvero una stringa che comincia con il carattere `\`, seguito da un OSC Type Tag String e da eventuali altri OSC Arguments. Un OSC Type Tag String è una stringa che comincia con il carattere `,` seguito da una serie di caratteri che rappresentano il tipo di dato OSC Arguments a cui corrispondono. Qualsiasi altra stringa contenente caratteri non riconosciuti dal protocollo OSC viene automaticamente scartata dal sistema. Una sequenza di OSC Arguments viene identificata dalla serie di rappresentazioni in codice binario di ogni singolo argomento.

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

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:

```
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 invia-

re 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.

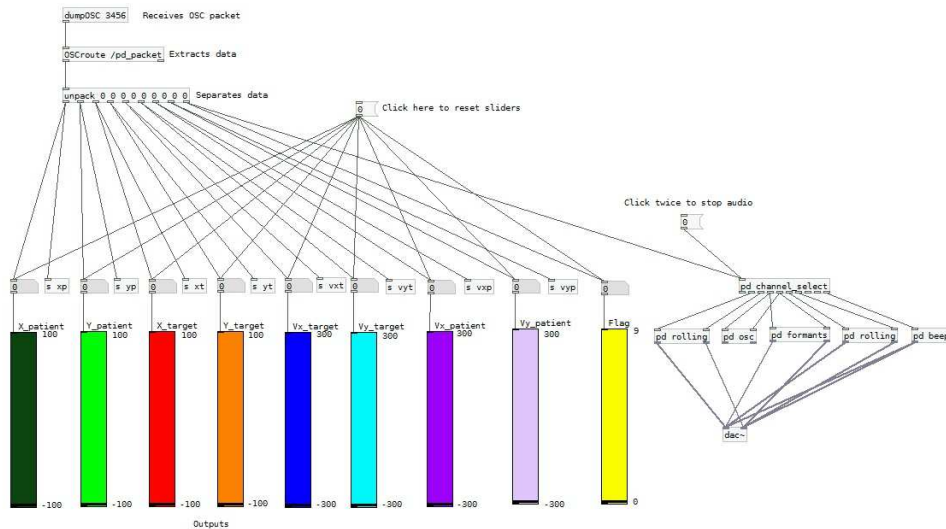


Figura 2.9: PureData: gli slider sulla sinistra rappresentano i valori ricevuti in ingresso, mentre i blocchetti sulla destra identificano i diversi tipi di feedback (pd rolling (1) = ATR, pd formats = APER, pd rolling (2) = AVER, pd beep = beep di inizio e fine esercizio)

2.6 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 tipo 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, '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.

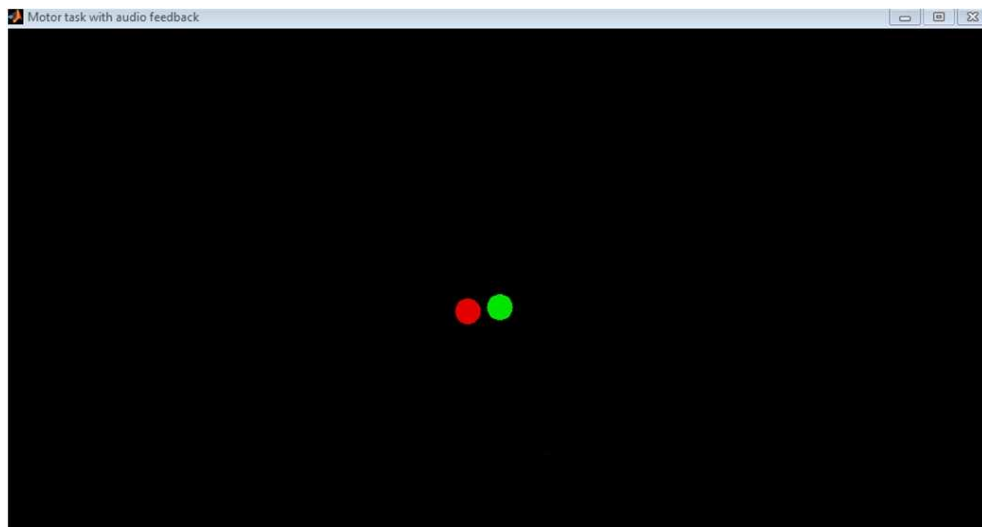


Figura 2.10: Finestra di esecuzione dell'esercizio con target (in rosso) e soggetto (in verde)

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 la tesi [49] [50], 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 [50]:

- 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).

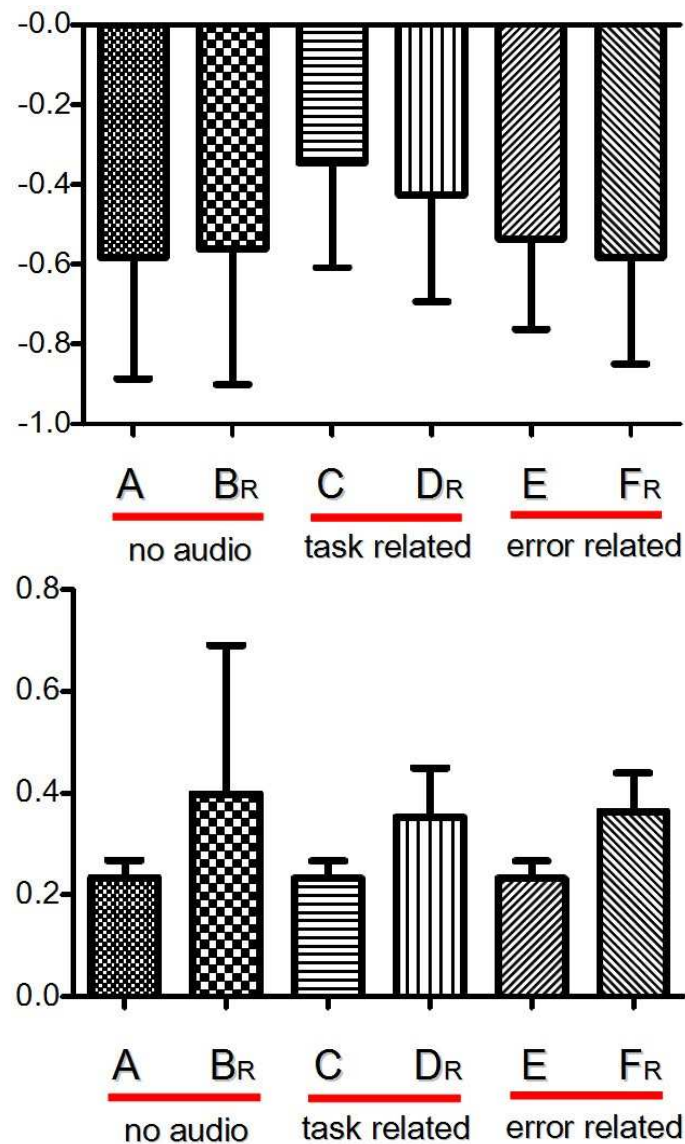


Figura 3.2: sopra: media pesata dell'errore di tracking (normalizzato rispetto al raggio del target); sotto: integrale della velocità relativa (normalizzato rispetto alla lunghezza del tratto). Task A (lunghezza fissa - no audio), B (lunghezza casuale - no audio), C (lunghezza fissa, audio task related), 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 feedback visivo e uditivo, i quali forniranno informazioni sulle stesse grandezze, seppur in 2 modalità differenti. Ci si aspetta che diverse modalità, visiva e sonora, siano più adatte a rappresentare grandezze diverse. Inoltre verrà implementato un nuovo tipo di feedback, legato all'errore di velocità del soggetto rispetto al target, il quale si spera possa dare un contributo per una maggiore accuratezza nell'inseguimento del target.

Il test completo è costituito da 7 task, diversi l'uno dall'altro per tipo di feedback e modalità con cui esso è stato fornito. Ogni task è costituito da una serie di movimenti di 'target tracking' orizzontali in direzione alternata destra/sinistra; l'esercizio inizia e termina a nella stessa posizione al centro dello schermo. Ogni task dura circa 90 secondi ed è preceduto da una fase iniziale, della durata di 5 secondi, che permette al soggetto di sovrapporsi al target fermo a centro schermo; tutti i task sono composti da 30 cicli: 28 di questi sono completi, in quanto il movimento interessa tutta la percentuale della larghezza dello schermo resa disponibile, mentre il primo e l'ultimo ciclo portano il target rispettivamente dalla posizione iniziale al primo estremo (ciclo 1) e dall'ultimo estremo nuovamente alla posizione iniziale (ciclo 30). La velocità media è stata impostata a 500 pixel/secondo per ogni task, la traiettoria, fissa, al 85% della larghezza dello schermo. La dimensione del raggio degli indicatori (il target è rappresentato da un pallino rosso, il soggetto da un pallino verde) è stata fissata a 25 pixel.

Sono stati implementati 7 tipi di task diversi:

1. Task A (NF): no feedback
2. Task B (CPER - color position error related): feedback visivo legato all'errore di posizione del paziente rispetto al target
3. Task C (CVER - color velocity error related): feedback visivo legato all'errore di velocità del paziente rispetto al target

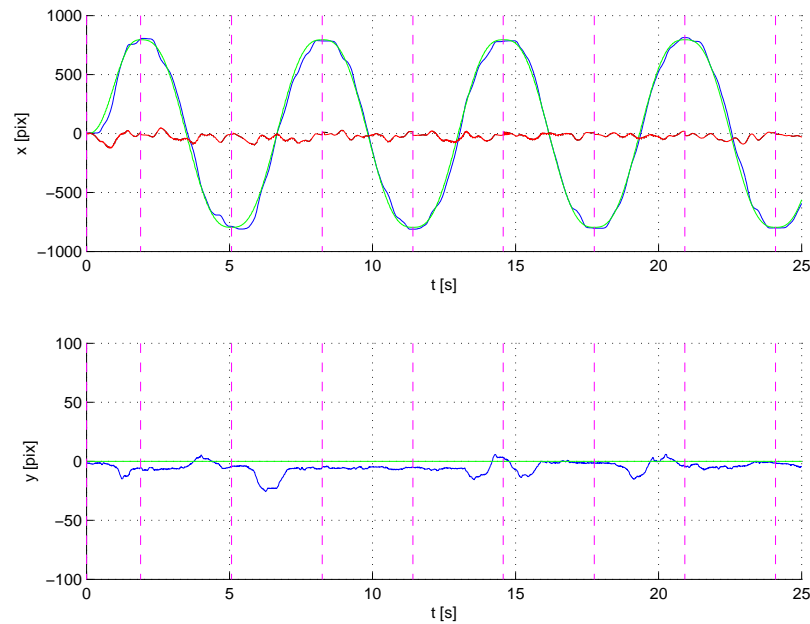


Figura 3.3: Grafici relativi alla performance di un soggetto. Sopra, grafico relativo alla posizione del soggetto (blu) e del target (verde) rispetto al tempo lungo l'asse x ; in rosso è visibile l'errore di posizione pari alla differenza $x_s - x_t$; i tratteggi viola indicano gli istanti di cambio ciclo, si noti che il primo ha una durata dimezzata rispetto agli altri. Sotto, grafico relativo alla posizione del soggetto (blu) e del target (verde) rispetto al tempo lungo l'asse y ; l'errore di posizione coincide con la posizione del soggetto stessa; i tratteggi viola indicano gli istanti di cambio ciclo.

4. Task D (CTR - color task related): feedback visivo legato alla velocità del target
5. Task E (APER - audio position task related): feedback uditivo spazializzato legato all'errore di posizione del paziente rispetto al target
6. Task F (AVER - audio velocity error related): feedback uditivo spazializzato legato all'errore di velocità del paziente rispetto al target
7. Task G (ATR - audio task related): feedback uditivo spazializzato legato alla velocità del target.

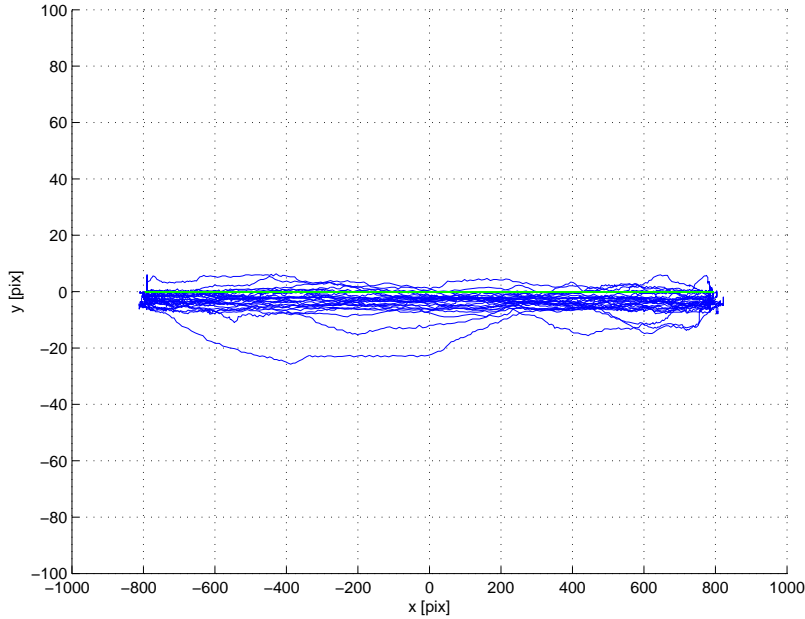


Figura 3.4: Grafico relativo al percorso del target (in verde) e del soggetto (in blu) lungo l'intero task (30 cicli) sull'asse x e y .

3.2.1 Feedback video

L'informazione visiva è percepita attraverso una variazione del colore dello sfondo del monitor: quando il feedback è legato ad un errore, sia di posizione che di velocità, lo schermo rimane nero in caso di errore nullo, mentre sfuma verso un colore azzurro proporzionalmente all'errore commesso.

L'errore di posizione è definito come:

$$\epsilon = \frac{\sqrt{(x_s(t) - x_t(t))^2 + (y_s(t) - y_t(t))^2}}{6R} \quad (3.1)$$

Tale formula considera l'errore di posizione come distanza sul piano (distanza euclidea) e pertanto tiene conto della differenza di coordinate sia in x che in y tra soggetto e target. Tale errore è stato normalizzato secondo un fattore pari a 6 volte il raggio degli indicatori.

L'errore di velocità è definito come:

$$\epsilon = \frac{(v_{x_s} - v_{x_t}) \text{sign}(x_s - x_t)}{1.5v_{tmax}} \quad (3.2)$$

dove:

- $v_{x_s} - v_{x_t}$: è la differenza tra la velocità del soggetto e quella del target
- $sign(x_s - x_t)$: è un fattore di correzione che rende negativo il numeratore quando la differenza di velocità tende a ridurre l'errore di posizione
- $1.5v_{t_{max}}$: è un fattore di normalizzazione

Successivamente è stata impostata una condizione per cui se $\epsilon < 0$ l'errore è considerato nullo: quindi, un soggetto in ritardo rispetto al target, anche se procede con velocità maggiore, non viene considerato in errore, poiché di fatto si sta avvicinando al target, cercando di migliorare la propria performance.

Quando il feedback è legato alla velocità del target, invece, lo schermo sfuma proporzionalmente a tale parametro: lo sfondo nero rappresenta la velocità nulla, mentre un azzurro sempre più acceso ne indica un aumento. Tale parametro è stato calcolato come:

$$\epsilon = \frac{|v_{t_x}|}{2v_{t_{max}}} \quad (3.3)$$

dove $2v_{t_{max}}$ è un fattore di normalizzazione pari a 2 volte la velocità massima del target.

Successivamente tutti i valori di ϵ sono stati elevati al cubo per rendere meglio l'idea di ciò che rappresentano, in questo modo vengono ridotti i piccoli errori e amplificati quelli grandi.

3.2.2 Feedback audio

Sono state implementate diverse tipologie di feedback audio:

- audio legato all'errore di posizione spazializzato
- audio legato all'errore di velocità spazializzato
- audio legato alla velocità del target spazializzato

Per la sintesi della prima modalità di audio 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:

$$\epsilon = |x_s - x_t| \quad (3.4)$$

L'audio legato all'errore di velocità, creato ex novo, è stato realizzato attraverso un suono che rievoca lo 'scratch' di un disco (dando comunque l'idea di un oggetto che rotola); esso va ad influenzare l'intensità del suono soltanto nella dimensione in x dello schermo ed è mappato sul volume e sulla tonalità; anche in questo caso gli ingressi sono stati normalizzati. L'errore di velocità è stato definito in analogia a quanto fatto per la modalità video, utilizzando la 3.2:

$$\epsilon = \frac{(v_{x_s} - v_{x_t}) \text{sign}(x_s - x_t)}{1.5v_{tmax}} \quad (3.5)$$

Per cui valgono le stesse considerazioni fatte con il corrispondente feedback visivo.

L'audio legato alla velocità del target riproduce un suono che simula quello del rotolamento di un oggetto su un piano: anche questo è stato ripreso dal lavoro fatto in precedenza [49] senza apportare nuove modifiche in quanto l'effetto ottenuto era già soddisfacente; il suono si integra perfettamente col movimento da simulare, 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.

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

Tabella 3.1: 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

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.

Convenzione sui tipi di segnali sonori

Per una maggiore comodità e chiarezza nello sviluppo dell'interfaccia è stata definita una convenzione sulla modalità con la quale riferirsi ai vari tipi di suoni utilizzati (tabella 3.1). Ad ogni segnale è stato univocamente associato un valore intero (audio_flag) con il quale è possibile identificarlo sia all'interno dell'interfaccia sia per la comunicazione con l'applicazione esterna deputata alla realizzazione dei suoni.

3.3 Popolazione

E' stata presa in considerazione una popolazione di 22 soggetti (età media di 23.48 anni, 18 uomini e 4 donne), privi di alcun tipo di disabilità motoria e gravi difetti agli apparati visivo e uditivo. Lo studio su soggetti sani è stato imposto da ragioni di comodità e da problemi di tempistica, i risultati ottenuti possono essere utilizzati per caratterizzare la normale risposta di un soggetto sano ad una determinata tipologia di feedback video/audio; questi possono rappresentare la base per confronti futuri con altre tipologie di soggetti ed, in particolare, possono essere anche estesi allo studio di pazienti colpiti da ictus.

Prima dell'esecuzione del test ad ogni soggetto è stato chiesto di compilare un questionario: questo è stato utile per capire se il soggetto fosse o meno idoneo a svolgere l'esercizio e per interpretare al meglio i risultati ottenuti. Sono state prese informazioni riguardanti:

- eventuali deficit della vista e/o dell'udito per avere la certezza che i feedback venissero colti appieno
- eventuali problemi di mobilità dell'arto impiegato nel test per non dover attribuire una cattiva performance a problemi preesistenti
- se si pratica uno sport e/o si usa di strumenti musicali e con che frequenza, per capire se il soggetto poteva aver sviluppato una sensibilità particolare dell'arto in questione
- le ore passate davanti allo schermo della tv o del pc per lavoro, navigazione, visione di film per capire quanto il soggetto è abituato a stare davanti a uno schermo
- eventuali ore di gioco col computer o con console per verificare se il soggetto poteva aver già acquisito una certa dimestichezza con un device

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 (17/22), quasi nessuno suona uno strumento musicale (3/22), i soggetti in media passano 2,14h al giorno davanti al pc e 4,13h al giorno davanti a 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.

3.3.1 Randomizzazione

Per rendere questo test il più attendibile e omogeneo possibile è stato scelto di randomizzare l'ordine di esecuzione dei task, in modo tale che per ogni soggetto la sequenza fosse diversa l'una dall'altra. Infatti appare lampante che, in proporzione, l'esecuzione del primo esercizio del test sarà sempre qualitativamente peggiore dell'esecuzione dell'ultimo. In particolare, si è fatto in modo che ogni soggetto dovesse affrontare alternativamente un feedback video e uno audio, così da evitare il rischio che potesse creare una dipendenza da una particolare modalità. Tra essi, in modo casuale, è stato inserito il task A (no feedback) perché ininfluenza rispetto all'esecuzione degli altri.

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.

L'obiettivo richiesto era semplicemente quello di seguire, al meglio delle possibilità, il movimento dell'indicatore circolare di colore rosso rappresentato a video. Il soggetto veniva istruito su postura e comportamenti corretti da tenere durante la fase di test; veniva, inoltre, informato della modalità (video o audio) e della tipologia (feedback legato ad un errore del soggetto o alla velocità del target) del feedback che gli sarebbero stati forniti. Prima dell'inizio del test, al soggetto veniva presentato un training di prova, necessario per prendere confidenza con il



Figura 3.5: Alcuni soggetti durante i test

joystick. Tra un task e il successivo era prevista una pausa di circa 30s.

Progressivamente all'esecuzione dei test, è stata compilata una tabella riassuntiva 3.2 contenente tutte le informazioni riguardanti ogni soggetto, la cronologia dei task, e 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

In fase di prova si sono riscontrati alcuni problemi per quanto riguarda la riproduzione del test con il feedback video: nel momento in cui lo schermo sfumava, il colore il video non restituiva più un'immagine fluida ma procedeva a scatti; inoltre, spesso accadeva che i pallini del target e del soggetto venissero in parte tagliati durante il movimento. Con queste premesse non si sarebbero potuti effettuare i test perché il feedback video sarebbe stato inutilizzabile. La causa è stata attribuita alla funzione *draw*, responsabile del cambio del colore di sfondo: l'ultima versione realizzata è risultata troppo pesante a discapito della renderizzazione del video. Si è preferito, quindi, riprendere una versione della funzione implementata precedentemente: è priva di alcune funzionalità, tra cui i 'beep' di inizio e fine esercizio, ma risulta più leggera e garantisce il regolare svolgimento del test.

Un ulteriore problema riscontrato per tempo, ovvero prima di sottoporre i soggetti al test, è inerente al fatto che spesso volte Windows era vittima di crash. Le ipotesi più plausibili per valutare la causa di tale effetto sono legate al fatto che il software Simulink nel generare un modello Real-Time assume la massima priorità per quanto riguarda le operazioni da svolgere. Gli altri programmi in esecuzione assumono una priorità praticamente nulla. Ecco perchè Matlab nel caricare i valori pervenuti dal modello Simulink non riesce a gestire la cosa con i tempi pilotati da Simulink stesso e si presume che questo porti ad un crash di Windows.

PIANIFICAZIONE TASK MOTORI									
DATA TEST	ORA TEST	GRUPPO	SOGGETTO	ETA'	SESSO	ORDINE TASK	BRACCIO	OCCHIALI/LENTI	IRREGOLARITA' NEI TEST
19/01/11	15.00 - 15.30	LD	Balsemin Antonio	22	M	F B G A D E C	DX	O	crash: task D
17/01/11	10.00 - 10.30	LD	Benedetti Giulio	24	M	B F D E C G A	DX	-	
18/01/11	15.00-15.30	LD	Biso Julian	21	M	G C F B E A D	DX	O	
18/01/11	17.30 - 18.00	LD	Casasola Alessandro	22	M	B A E C F D G	DX	-	
17/01/11	17.00 - 17.30	LD	Calore Andrea Roberto	22	M	E C G D F B A	DX	O	
19/01/11	14.00 - 14.30	LD	Carraro Alessandro	22	M	C G A B E D F	DX	O	
17/01/11	17.30 - 18.00	LD	De Faveri Valentino	24	M	B G A D E C F	DX	-	crash: task A
17/01/11	10.30 - 11.00	LD	Dermati Damiano	27	M	F D A G C E B	DX	-	
17/01/11	9.30 - 10.00	LD	Gazaneo Andrea	23	M	G D A E B F C	DX	L	
19/01/11	11.00 - 11.20	LD	Guggia Silvia	22	F	F D G B E A C	DX	O	
17/01/11	15.00-15.30	LD	Lorenzin Aldo	25	M	B E C F A D G	DX	O	
18/01/11	10.00 - 10.30	LD	Maroni Ida	22	F	A C G D E B F	DX	-	task E ripetuto joystick pos. errata
19/01/11	14.30 - 15.00	LD	Moressa Matteo	22	M	G A C F B E D	DX	-	crash: task D
18/01/11	10.30-11.00	LD	Nucibella Enrico	27	M	F C G A D E B	DX	O	trapano + martello
17/01/11	12.30 - 13.00	LD	Nunes Marco	22	M	A E D G C F B	DX	-	
18/01/11	16.00 - 16.30	LD	Papale Davide	23	M	C E D A G B F	DX	O	
19/01/11	16.00-16.30	LD	Pellegrini Luca	22	M	B E C A G D F	DX	-	
18/01/11	16.30 - 17.00	LD	Romani Daniele	25	M	F D E C G B A	DX	-	
17/01/11	12.00 - 12.30	LD	Rosada Marco	22	M	D G B E A C F	DX	-	confonde tgt col p. verde (10-13)
17/01/11	15.30 - 16.00	LD	Savarino Federica	22	F	A F C G B E D	DX	O	
17/01/11	11.00 - 11.30	LD	Stefani Anna	22	F	B E D G C A F	DX	L	
17/01/11	11.30 - 12.00	LD	Zorzi Andrea	23	M	F A C E B G D	SX	-	

Tabella 3.2: Tabella relativa alle informazioni sui soggetti partecipanti e note prese durante l'esecuzione dei task.

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. Dal momento che il risultato che si vuole evidenziare sono le modifiche apportate dalla presenza di un appropriato feedback negli inseguimenti, non si andrà mai ad effettuare un'analisi per gruppi, ma si andranno invece a valutare sempre le differenze che sussistono fra l'esecuzione di due task differenti (per tipologia di feedback) eseguiti dallo stesso soggetto. Per consentire l'analisi statistica si andrà a calcolare, per ogni task di ogni soggetto, una serie di parametri i cui valori daranno un'indicazione, più o meno globale, della 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 (0,01s). Il calcolo di questo parametro di errore consiste nell'implementazione del metodo dei rettangoli per il calcolo numerico dell'integrale:

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

dove N rappresenta il numero di campioni del ciclo, mentre il fattore di normalizzazione presente al denominatore rappresenta la lunghezza del tratto percorso nel ciclo considerato.

Tale parametro misura lo spazio percorso dal soggetto avente modulo del vettore velocità pari a $v_s - v_t$ e pertanto 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:

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

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

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

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

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

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

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

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

con

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

dove:

- $(v_{x_s}(t) - v_{x_t}(t))\text{sign}(x_s(t) - x_t(t)) < 0$: stabilisce se il soggetto è in ritardo rispetto al target nell'istante in cui quest'ultimo cambia ciclo
- $v_{x_s}(t) \neq v_{x_t}(t)$: verifica, in base ai segni discordi delle velocità, che i 2 indicatori abbiano direzioni opposte
- $|v_{x_t}(t) - v_{x_t}(t - t_c)| > 0$: 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.

4.1.4 Errore medio di posizione in y (emy)

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

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

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

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

4.1.5 Errore quadratico medio in x (eqmx)

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

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

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

4.1.6 Errore quadratico medio in y (eqmy)

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

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

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

4.1.7 Lead error (lead)

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

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

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

4.1.8 Lag error (lag)

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

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

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

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. Dal momento che tutti i soggetti hanno eseguito tutti i task è stato possibile effettuare un'analisi per dati appaiati, confrontando le prestazioni dello stesso soggetto fra i vari task. Il limite di significatività del p-value è stato impostato a 0,05. Per l'analisi sono stati scelti il Wilcoxon test e il t test:

4.2.1 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, con il cui termine si fa riferimento alla differenza tra ogni valore e la mediana. Il risultato del Wilcoxon test hanno senso solo se la dispersione è casuale, cioè che un fattore che causa un valore troppo alto o troppo basso interessi solo per quel dato valore.

4.2.2 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. L'opzione 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 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 interessa solo per quel dato valore.

Avendo notato che la distribuzione dei campioni non è una distribuzione gaussiana, si è preferito utilizzare il Wilcoxon test a t test, sebbene, comunque, siano stati effettuati entrambi.

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 22 soggetti totali ne sono stati scartati 4. 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: per questo motivo è stato scelto un intervallo compreso tra il ciclo

2 e 29 compresi. A partire sempre 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: essa è stata effettuata su 6 soggetti, in totale, sono stati modificati 8 task.

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 va a cambiare i dati dalla tabella di origine. Sono stati generati 8 file.pzf relativi alle 8 misure calcolate. All'interno del programma sono stati impostati tutti i confronti tra task desiderati da effettuare sia con il Wilcoxon che con il t test; ogni file 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.

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. *updateEx.m* elimina i valori salvati oltre l'istante *indmax* (istante di fine esercizio) e rielabora tutti i parametri relativi alle posizione del target e del soggetto togliendo un offset pari a metà delle dimensioni dello schermo, sia lungo l'asse x che l'asse y: in tal modo l'origine degli assi di riferimento viene riposizionata a centro schermo (da effettuare solo alla prima elaborazione)

The screenshot shows the GraphPad Prism software interface. The main window displays a data table with the following structure:

		Task A		Task B		Task C		Task D		Task E		
		Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	
1	BalseminAntonio	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	BisoJulian	0.859600	0.630010	1.295700	0.939840	1.425800	1.030700	1.447900	1.184600	0.869510	0.605630	0.877
4	CaloreAndreaRo	0.891480	0.585640	1.814100	1.555000	1.376300	0.991160	1.229400	0.962260	0.894500	0.648490	0.867
5	CarraroAlessand	0.822470	0.579740	1.483400	1.192900	1.666300	1.274500	1.177700	0.918450	1.176200	0.663640	0.824
6	CasasolaAlessa	1.132300	0.806940	1.792300	1.522200	1.524100	1.078400	1.784100	1.422300	1.088700	0.871360	1.074
7	DeFaveriValent	1.160000	0.806980	2.035700	1.567100	1.399700	0.998690	1.899000	1.441500	1.020600	0.727060	0.984
8	DermatiDamiano	1.238000	0.858850	3.179700	2.051500	2.100100	1.653000	2.058500	1.379400	1.759300	1.225300	1.178
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	LorenziniAldo	1.296700	1.040400	2.504800	1.753000	1.806500	1.412900	1.983800	1.535800	1.684400	1.203000	1.168
12	Maronilda	0.817210	0.606650	1.578200	1.387800	1.175900	0.886810	1.473100	1.203100	0.899680	0.874770	0.718
13	MoressaMatteo	0.908110	0.682900	1.342600	0.839300	1.226600	0.862750	1.095900	0.742720	0.951030	0.708580	0.772
14	NucibellaEnrico	1.144300	0.897470	1.224400	1.061600	1.326500	0.911710	1.601100	1.271700	1.008900	0.663580	1.048
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	1.098300	0.794300	0.774450	0.563210	0.664
17	PellegriniLuca	0.891080	0.699360	1.435200	1.032100	1.253200	0.923240	1.226300	0.837400	1.010600	0.797930	0.970
18	RomaniDaniele	0.811580	0.574360	1.237800	0.854940	1.620200	1.258900	1.824000	1.274700	0.836840	0.560570	0.883
19	RosandMarco	1.329700	0.826730	1.923400	1.229700	1.912800	1.283800	3.436400	1.906700	1.287400	0.822020	1.218
20	SavainoFederica	1.717400	1.170900	2.747500	2.449000	1.821600	1.410100	2.249900	1.774600	1.145400	0.835100	1.524
21	StefaniAnna	1.396800	0.927430	2.241300	2.056000	1.790000	1.293900	2.706400	2.491400	1.340000	0.896930	1.178
22	ZorziAndrea	0.930420	0.730450	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 4.1: Interfaccia grafica di GraphPad Prism

2. *saveDat.m* carica i file *.mat* aggiornati dall'update e calcola i parametri utili ad una prima elaborazione, salvando i risultati in un corrispondente file *.dat*. In particolare, per ogni task, vengono calcolati tutti i parametri definiti nel paragrafo precedente: errore di posizione, integrale del modulo della velocità, ritardo medio (errore di posizione lungo l'asse x), errore di posizione lungo l'asse y, errore quadratico medio in x, errore quadratico medio in y, 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*
3. *saveDatTaroc.m* elimina i cicli anomali attraverso il procedimento spiegato sopra; 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
4. *xlsDatColor.m* crea i file *.xls*, uno per ogni parametro, contenenti media e deviazione standard di ciascun soggetto riferite ad ogni task; quindi, in

totale, vengono generate 8 tabelle nelle cui colonne sono contenuti i valori di media e deviazione standard dei 7 task, mentre nelle righe i valori di ogni singolo soggetto in tutti i task. Viene inoltre generato il file `settings.mat` per tenere traccia dei parametri impostati al lancio del file

5. *pScript.pzc*, implementato in Prism, importa i valori dei p-value generati da ogni confronto nei due file `W.csv` e `t.csv`, rispettivamente del Wilcoxon e del t test; genera inoltre il file `p-values-color.xlsx` che contiene il riferimento agli altri due
6. i tre file devono essere aperti contemporaneamente: il file `p-values-color.xlsx` contiene due fogli che vengono aggiornati automaticamente prendendo i contenuti di `W.csv` e `t.csv`, pertanto è necessario salvarlo prima di ogni chiusura
7. *saveResColor.m* salva in una sottocartella della cartella ‘confronti’, infine, tutti i grafici di Prism e le tabelle Excel di ogni parametro, il file `settings.mat` e il file.xlsx del p-value

4.4 Risultati

In questo paragrafo vengono presentati i principali risultati ottenuti dall’elaborazione dei dati. Sono state confrontate le seguenti coppie di task:

- modalità no feedback con gli altri 6 task: A-B, A-C, A-D, A-E, A-F, A-G
- feedback video e audio: B-E, C-F, D-G
- feedback video: B-C, C-D, B-D
- feedback audio: E-F, F-G, E-G

Si è deciso di limitare la descrizione dei risultati solo all’interpretazione di alcuni parametri ritenuti più significativi: `dtf`, `distm`, `emx`. Di seguito vengono riportati i grafici relativi 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).

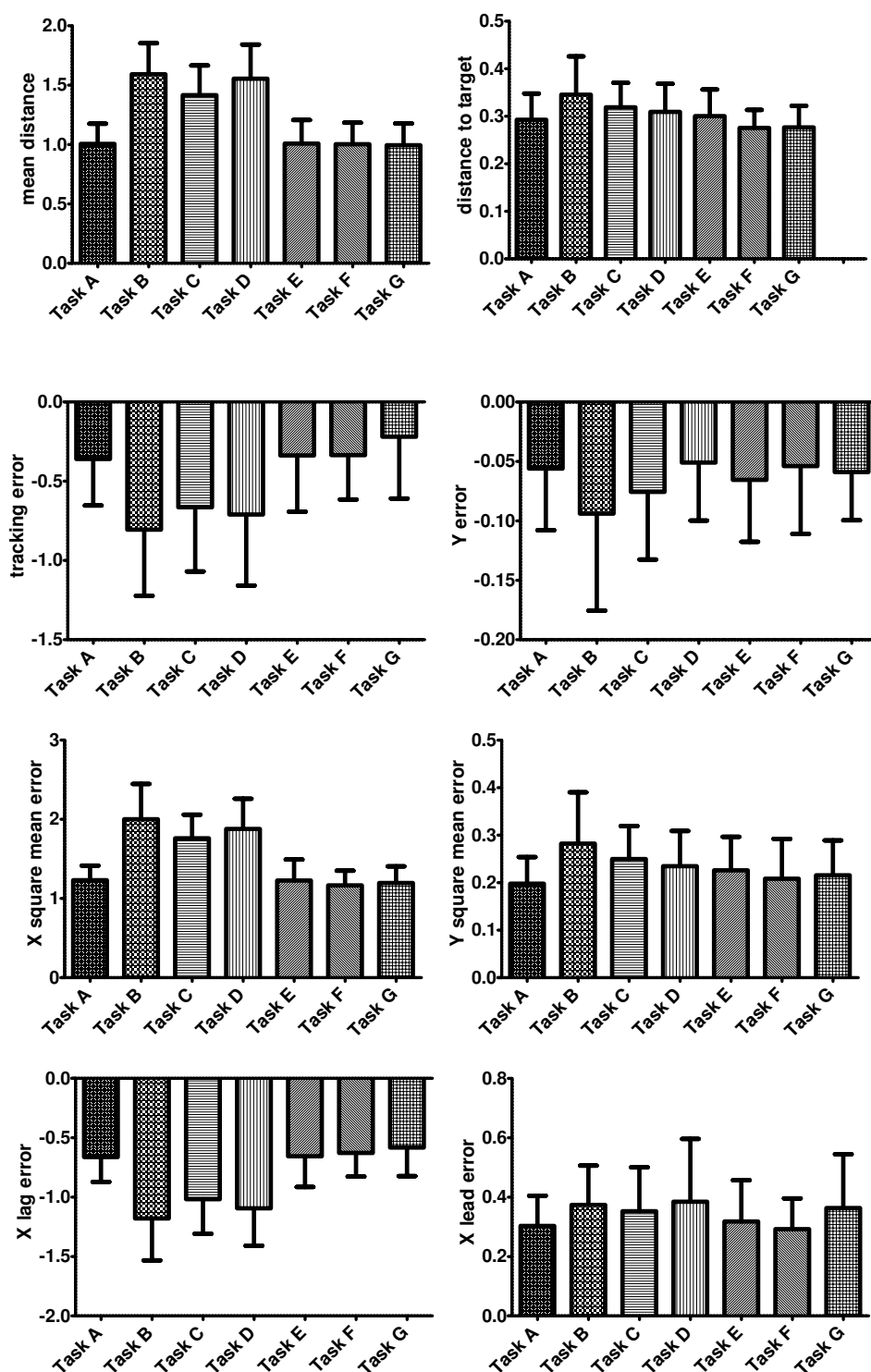


Figura 4.2: Grafici relativi ai parametri considerati nell'elaborazione: mean distance (dism), distance to target (dtt), tracking error (emx), Y error (emy), X square mean error (eqmx), Y square mean error (eqmy), X lag error (lag), X lead error (lead). Tutte le grandezze sono state normalizzate rispetto al raggio del target tranne il dtt, normalizzato rispetto alla lunghezza del tratto percorso.

P-VALUE color (Paired t-test)															
	A-B	A-C	A-D	A-E	A-F	A-G	B-C	C-D	B-D	E-F	F-G	E-G	B-E	C-F	D-G
distm	< 0.0001	< 0.0001	< 0.0001	0.6641	0.1532	0.6911	0.0094	0.0841	0.3163	0.1693	0.4207	0.4835	< 0.0001	< 0.0001	< 0.0001
dtt	0.0070	0.0676	0.1649	0.4461	0.1643	0.2088	0.1109	0.4631	0.0664	0.0510	0.9138	0.0623	0.0020	0.0032	0.0224
emx	< 0.0001	< 0.0001	< 0.0001	0.6380	0.5418	0.0298	0.1373	0.3745	0.3701	0.9744	0.0638	0.1341	< 0.0001	< 0.0001	< 0.0001
emy	0.0445	0.0967	0.6824	0.2687	0.8159	0.6537	0.4063	0.0414	0.0279	0.2293	0.6037	0.4021	0.1155	0.1216	0.4737
eqmx	< 0.0001	< 0.0001	< 0.0001	0.9422	0.0767	0.4612	0.0119	0.1138	0.3072	0.1985	0.4384	0.5462	< 0.0001	< 0.0001	< 0.0001
eqmy	0.0008	0.0126	0.0040	0.0279	0.4592	0.2167	0.2826	0.4426	0.0280	0.2869	0.6365	0.3927	0.0058	0.0848	0.1180
lag	< 0.0001	< 0.0001	< 0.0001	0.8230	0.0725	0.0108	0.0242	0.0238	0.2820	0.3865	0.1420	0.1238	< 0.0001	< 0.0001	< 0.0001
lead	0.0311	0.1314	0.0865	0.5423	0.6999	0.1434	0.6223	0.4562	0.8479	0.4514	0.0665	0.2790	0.0939	0.0839	0.6892

P-VALUE color (Wilcoxon)															
	A-B	A-C	A-D	A-E	A-F	A-G	B-C	C-D	B-D	E-F	F-G	E-G	B-E	C-F	D-G
var-distm	0.0002	0.0002	0.0002	0.8276	0.1507	0.7939	0.0166	0.0894	0.3380	0.2066	0.5713	0.7605	0.0002	0.0002	0.0002
var-dtt	0.0131	0.0674	0.2227	0.5136	0.2397	0.4080	0.2575	0.4080	0.0815	0.0451	0.8961	0.1071	0.0090	0.0090	0.0264
var-emx	0.0003	0.0003	0.0004	0.6632	0.6951	0.0235	0.0235	0.2959	0.3838	0.8276	0.0742	0.1507	0.0002	0.0003	0.0003
var-emy	0.0553	0.1274	0.9306	0.2575	0.8617	0.6319	0.4859	0.0407	0.0264	0.2227	0.5136	0.4591	0.0742	0.1387	0.5136
var-eqmx	0.0002	0.0002	0.0002	0.8276	0.1071	0.6632	0.0147	0.1274	0.4080	0.2575	0.5421	0.8617	0.0002	0.0002	0.0002
var-eqmy	0.0003	0.0061	0.0079	0.0329	0.5421	0.1634	0.1770	0.6319	0.0166	0.2959	0.3165	0.4591	0.0041	0.0979	0.2066
var-lag	0.0002	0.0002	0.0002	0.4080	0.0611	0.0115	0.0147	0.0329	0.3838	0.6013	0.1507	0.1507	0.0002	0.0002	0.0002
var-lead	0.0366	0.1507	0.1169	0.7275	0.4331	0.2959	0.7605	0.8961	0.5713	0.5713	0.1071	0.4859	0.0742	0.0979	0.4859

Figura 4.3: Tabella dei p-value: in azzurro i confronti significativi ($<0,05$), in rosa confronti che sono quasi significativi ($<0,1$).

- uno degli scopi di questo lavoro era quello di effettuare un confronto tra le diverse modalità con cui fornire i vari tipi di feedback, pertanto è stato eseguito un confronto diretto tra la modalità video e quella audio. Come mostrano le figure, e come conferma la tabella del p-value, è emerso che l'utilizzo del colore di sfondo non ha per niente aiutato lo svolgimento dei task: in particolare, i confronti diretti tra B-E, C-F, D-G godono della significatività in tutti e tre i parametri di valutazione, dove la differenza sta ad indicare che la modalità audio è nettamente migliore. Addirittura, le performance nei task B,C,D risultano peggiori rispetto a quelle non assistite da alcun feedback, soprattutto per quanto riguarda i parametri d_{stm} e emx . Ciò potrebbe essere interpretato con il fatto che la modalità visiva viene sovraccaricata in quanto l'informazione relativa al colore di sfondo si sovrappone alla visione degli indicatori
- questa ipotesi è confermata anche dal fatto che, confrontando le tre modalità video tra di loro, il task C risulta essere migliore rispetto al B e al D: questo accade perché il feedback utilizzato in questa modalità è meno sensibile all'errore commesso, quindi si verificano poche variazioni di colore dello sfondo che distraggono di meno il soggetto. Nelle altre due modalità il feedback o è più sensibile all'errore commesso (task B) o è continuo (task D), questo ha portato un peggioramento delle performance da parte dei soggetti
- rispetto alla modalità NF, ATR (task G, audio legato alla velocità del target) ha l'effetto di ridurre il ritardo medio (emx), dovuto alla contemporanea riduzione del lag error e aumento del lead error: questo fatto è confermato dalla significatività del p-value nel confronto A-G rispetto a tale parametro. Non si verifica, invece, una riduzione delle ampiezze dei movimenti correttivi attorno al target (d_{tt}), il che porta a concludere che il feedback uditivo relativo alla velocità del target ha come principale effetto quello di traslare nella direzione di anticipo il range di variabilità dei movimenti di correzione. Queste conclusioni, inoltre, confermano i risultati ottenuti nei test precedenti [49]

- il feedback legato all'errore di velocità (Task F) non ha portato significativi vantaggi in termini di performance: come dimostrano i grafici 4.2 e le tabelle dei p-value 4.3, risulta peggiore rispetto al task G; tuttavia, è paragonabile al feedback relativo all'errore di posizione (task E) ad eccezione per il dtt, in cui risulta più efficace.

4.4.1 Effetto del learning

A seguito di queste prime analisi si è indagato più a fondo ed, in particolare, si è andato a considerare un dettaglio fin qui tralasciato: tutti i soggetti, seppur in ordine casuale, hanno svolto tutti e 7 i task uno di seguito all'altro con una pausa di 30 secondi; da questo si potrebbe ipotizzare che dal primo all'ultimo task eseguito ci possa essere stato un effetto di apprendimento dell'esercizio, *learning*. Poiché si è scelta una traiettoria a lunghezza fissa l'esercizio diventava ripetitivo nel movimento.

Una possibile conseguenza di questo fatto potrebbe essere un esito falsato dei risultati: infatti, qualora la randomizzazione avesse concentrato in un numero considerevole di sequenze lo stesso task nella stessa posizione di esecuzione, i valori calcolati per questo task sarebbero potuti essere migliori o peggiori a seconda che si trovasse spesso alla fine o all'inizio delle sequenze random. A questo proposito si è deciso di fare un'attenta analisi dei dati considerando l'ordine di esecuzione dei task di ogni soggetti e andandoli a modificare applicando un certo fattore correttivo per ovviare al possibile effetto di learning. Per fare ciò si è ripreso il file *xlsDatColor.m* al cui interno è stata aggiunta una parte di codice utile appunto per la modifica.

```
% correzione per learning
if learning
    % caricamento dati di permutazione task
    [A,B] = xlsread([resdirname 'confronti\RANDOM.xlsx']);
    indSubj = zeros(1,length(subjNames));
    for ns = 1:length(subjNames)      % per ogni soggetto
        indSubj(ns) = find(strcmp(B,subjNames(ns)));
    end
    perm = A(indSubj,:);

    % calcolo dei fattori correttivi in base alla regressione
```

```

Mdttm = Mdt(:,1:2:end-1);
Mdttm = Mdttm./repmat(mean(Mdttm),length(subjNames),1);
dttperm = zeros(size(Mdttm));
pSubj = zeros(length(subjNames),2);
fattc = zeros(size(Mdttm));
fattcperm = zeros(size(Mdttm));
for ns = 1:length(subjNames) % per ogni soggetto
    dttperm(ns,:) = Mdttm(ns,perm(ns,:));
    pSubj(ns,:) = polyfit(1:size(Mdttm,2),dttperm(ns,:),1);
    fattc(ns,:) = 1./(1+[-3:3]*pSubj(ns,1));
    fattcperm(ns,perm(ns,:)) = fattc(ns,:);
end
Mfattcperm = zeros(size(Mdt));
Mfattcperm(:,1:2:end-1) = fattcperm;
Mfattcperm(:,2:2:end) = fattcperm;

% correzione dati per learning
for i=1:length(varNames) % per ogni variabile
    eval(['M' varNames{i} '=M' varNames{i} '.*Mfattcperm;'])
end
end

```

Per la stima dell'effetto di apprendimento si è deciso di prendere in considerazione il parametro dtt in quanto rappresenta al meglio l'andamento dei soggetti nel tempo, quindi si è proceduto calcolando una matrice di coefficienti correttivi per ogni soggetto, specifico per ogni task, da applicare a tutti i parametri dell'elaborazione:

1. per prima cosa si è sfruttato un file.xlsx contenente l'elenco di tutti i soggetti in ordine alfabetico e le rispettive randomizzazioni in modo da poter effettuare un confronto con i soggetti considerati per l'analisi: così facendo, si possono associare correttamente i nomi alle relative randomizzazioni ed escludere le randomizzazioni dei soggetti scartati dall'elaborazione
2. viene calcolato il rapporto tra il dtt di ogni soggetto e il dtt medio del gruppo per ogni task e quindi caricato in una matrice
3. per ogni soggetto, (quindi per ogni riga della matrice appena calcolata), si permutano tutti i task (colonne della matrice) in modo da riordinarli secondo l'ordine di esecuzione

4. si procede, quindi, con il calcolo della regressione lineare dei dati: viene stimata la pendenza percentuale (pp) che indica il grado di miglioramento della performance nel passaggio da una prova alla successiva
5. viene calcolato così il fattore di correzione per ciascuna prova:

$$fc = \frac{1}{1 + kpp} \quad (4.13)$$

dove k indica la distanza (con segno) del task considerato da quello centrale ($k=-3, -2, -1, 0, 1, 2, 3$)

6. la matrice di fattori correttivi personalizzati viene quindi permutata in base alle randomizzazioni in modo da riavere tutti i valori riordinati secondo i task (da A a G)
7. tale matrice viene moltiplicata per tutte le matrici contenenti i parametri utilizzati per l'elaborazione in modo da applicare una correzione generale, non solo al dtt

Il risultato di tutto questo procedimento ha fatto sì che i dati relativi ai primi tre task venissero ridotti gradualmente, il quarto rimanesse invariato perché il più veritiero, gli ultimi tre amplificati sempre in proporzione alla posizione assunta. La correzione analoga è stata ovviamente applicata anche al calcolo di tutte le deviazioni standard.

Basandoci sul calcolo della pendenza percentuale pp si può affermare l'effettiva esistenza di un effetto di learning: la pp rappresenta il coefficiente angolare (percentuale) della retta di regressione, il fatto che il suo valore sia risultato negativo per tutti i soggetti sta proprio ad indicare che la distanza media dal target è diminuita linearmente man mano che si procedeva con lo svolgimento dei task. In più, è stata calcolata la media tra tutti i soggetti della pendenza, ottenendo come risultato $pp = -4,5\%$ con un range che varia da $-6,6\%$ a $-1,1\%$, ciò significa che mediamente si è riscontrato un miglioramento del $4,5\%$ da ciascuna prova alla successiva.

Dopo queste prime considerazioni si è proceduto all'analisi statistica dei dati così elaborati e si è fatto un confronto con quelli ottenuti in precedenza: se si

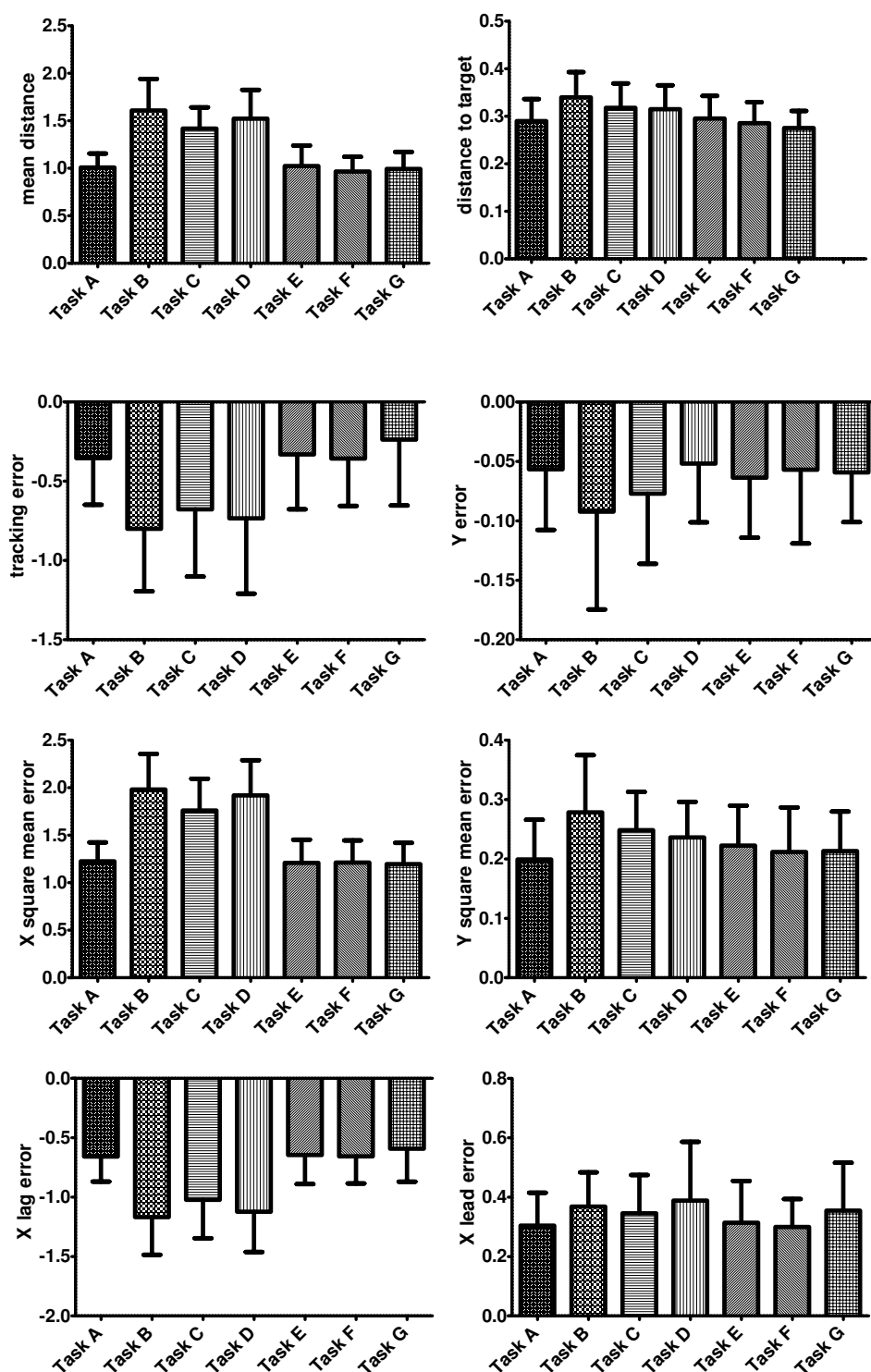


Figura 4.4: Grafici relativi ai parametri considerati nell'elaborazione con correzione in base al learnign calcolato: mean distance (dism), distance to target (dtt), tracking error (emx), Y error (emy), X square mean error (eqmx), Y square mean error (eqmy), X lag error (lag), X lead (lead). Tutte le grandezze sono state normalizzate rispetto al raggio del target tranne il dtt, normalizzato rispetto alla lunghezza del tratto percorso.

P-VALUE color (Paired t-test) learning															
	A-B	A-C	A-D	A-E	A-F	A-G	B-C	C-D	B-D	E-F	F-G	E-G	B-E	C-F	D-G
distrm	<0.0001	<0.0001	<0.0001	0.9476	0.9323	0.7502	0.0150	0.0089	0.6125	0.8793	0.8164	0.7221	<0.0001	<0.0001	<0.0001
dtm	<0.0001	0.0011	0.0012	0.3904	0.5058	0.0369	0.0146	0.7408	0.0297	0.1558	0.0960	0.0027	<0.0001	0.0002	<0.0001
emx	<0.0001	<0.0001	<0.0001	0.6146	0.9561	0.0953	0.2450	0.3728	0.5644	0.6645	0.0894	0.2485	<0.0001	<0.0001	<0.0001
emy	0.0562	0.0821	0.7151	0.4034	0.9730	0.7052	0.4738	0.0468	0.0463	0.4724	0.8519	0.5775	0.1133	0.1426	0.5221
eqmx	<0.0001	<0.0001	<0.0001	0.7166	0.7625	0.5396	0.0345	0.0279	0.5565	0.9471	0.7429	0.8024	<0.0001	<0.0001	<0.0001
eqmy	0.0002	0.0061	0.0012	0.0603	0.1893	0.1852	0.1909	0.3741	0.0249	0.3452	0.8461	0.3283	0.0015	0.0301	0.0248
lag	<0.0001	<0.0001	<0.0001	0.6691	0.9554	0.1293	0.0880	0.0288	0.5821	0.7600	0.1769	0.2987	<0.0001	<0.0001	<0.0001
lead	0.0068	0.1565	0.0595	0.7168	0.8271	0.1433	0.4568	0.2494	0.6812	0.6372	0.0690	0.2920	0.1011	0.1032	0.4970

P-VALUE color (Wilcoxon) learning															
	A-B	A-C	A-D	A-E	A-F	A-G	B-C	C-D	B-D	E-F	F-G	E-G	B-E	C-F	D-G
var-distrm	0.0002	0.0002	0.0002	0.7275	0.8276	1	0.0294	0.0115	0.5421	0.9653	0.8617	0.6951	0.0002	0.0002	0.0002
var-dtm	0.0003	0.0027	0.0023	0.3838	0.4331	0.0742	0.0166	0.8276	0.0366	0.1507	0.1071	0.0061	0.0006	0.0006	0.0006
var-emx	0.0004	0.0004	0.0005	0.7939	0.8617	0.0294	0.0894	0.3604	0.6013	1	0.1071	0.2397	0.0002	0.0006	0.0003
var-emy	0.0979	0.0979	0.8276	0.3838	0.9653	0.6951	0.5421	0.0451	0.0500	0.5136	0.6319	0.8617	0.1071	0.0979	0.5421
var-eqmx	0.0002	0.0002	0.0002	0.6632	0.6319	0.5713	0.0329	0.0329	0.4859	1	0.7939	0.7605	0.0002	0.0002	0.0002
var-eqmy	0.0003	0.0069	0.0041	0.0979	0.2227	0.1770	0.2959	0.6632	0.0264	0.3380	0.9306	0.1770	0.0020	0.0611	0.0407
var-lag	0.0002	0.0002	0.0002	0.4591	0.9306	0.0235	0.0407	0.0500	0.5713	0.8617	0.1770	0.2397	0.0002	0.0002	0.0002
var-lead	0.0069	0.2066	0.0742	0.7939	0.8276	0.2397	0.6319	0.2066	0.8617	0.9306	0.1071	0.3165	0.0131	0.1274	0.5136

Figura 4.5: Tabella dei p-value con correzione del learning: in azzurro i confronti significativi (<0,05), in rosa confronti che sono quasi significativi (<0,1).

osservano i grafici dei p-value 4.5 delle due elaborazioni si nota che non vi è una sostanziale differenza. Si può notare, però, che nell'elaborazione precedente risultavano significativi alcuni confronti che ora non lo sono più (dtt: E-F; emx: B-C), ma anche viceversa (dtt: A-C, A-D, B-C).

Il fatto che non si sia verificato uno stravolgimento dei risultati non significa che il calcolo del 'learning' sia stato inutile, ma che l'ordine con cui sono stati effettuati i test è stato ben randomizzato, cioè che tutti e 7 i task hanno occupato, con la stessa frequenza, tutte le posizioni possibili nell'ordine di svolgimento.

L'unica eccezione risulta essere il task F: dai nuovi grafici 4.4 appare, infatti, che sia stato svolto leggermente peggio (in particolare per quanto riguarda il dtt) di quanto non risulti dai dati originali. Da un'attenta analisi delle randomizzazioni si è notato che il task in questione è presente in modo più frequente nell'ultima parte della sequenza. Ciò significa che i dati originali di questo task hanno subito una maggiore influenza dal learning (in positivo) rispetto ai dati degli altri task.

Conclusioni

I risultati ottenuti hanno confermato l'importanza del feedback audio per gli esercizi di riabilitazione: l'informazione sonora apre, infatti, un altro canale di comunicazione attraverso il quale vengono fornite informazioni complementari a quelle già disponibili tramite la vista; in particolare, il feedback audio relativo alla velocità del target, informazione non valutabile precisamente attraverso il solo canale visivo, si è rivelato essere molto più adatto rispetto a quelli legati all'errore di posizione o di velocità. È emerso, invece, che la modalità video non ha per niente aiutato lo svolgimento dei task, anzi, in alcuni casi le performance risultano essere addirittura peggiori rispetto a quelle non assistite da feedback. Ciò potrebbe essere interpretato con il fatto che la modalità visiva viene sovraccaricata in quanto l'informazione relativa al colore di sfondo si sovrappone alla visione degli indicatori; non si esclude, tuttavia, che i cattivi risultati dipendano anche da una cattiva progettazione del feedback.

Per i progetti futuri, sarebbe interessante sviluppare nuovi modi di rappresentazione del feedback video, il quale è un ottimo strumento in sede di riabilitazione; si potrebbero inoltre estendere i test a soggetti malati così da poter capire se i feedback sviluppati siano davvero uno strumento efficace per un migliore e più pronto recupero della capacità motoria.

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] K. Mononen, "The effect of augmented feedback on motor skill learning in shooting," Ph.D. dissertation, Department of Health Sciences, University of Jyväskylä, 2007.
- [49] M. Battiston, "Studio dell'utilizzo del feedback uditivo nell'esecuzione di task motori," Master's thesis, Università degli studi di Padova, 2010.
- [50] 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.