

UNIVERSITÀ DEGLI STUDI DI PADOVA
FACOLTÀ DI INGEGNERIA

—
DIPARTIMENTO DI INNOVAZIONE MECCANICA E GESTIONALE
—

TESI DI LAUREA TRIENNALE IN INGEGNERIA BIOMEDICA

STUDIO DELL'UTILIZZO DI
FEEDBACK Uditivo
NELL'ESECUZIONE DI TASK
MOTORI

RELATORE: CH.MO PROF. ING. GIULIO ROSATI

LAUREANDO: MARCO BATTISTON

ANNO ACCADEMICO 2009-2010

Indice

Sommario	VII
Introduzione	IX
1 Feedback sonoro nella neuroriabilitazione	1
1.1 Obiettivi	3
1.2 Software di elaborazione dei dati	3
1.3 Test su soggetti sani	4
2 Software sviluppato	5
2.1 Schema di funzionamento della GUI	5
2.2 Creazione e visualizzazione dell'esercizio	7
2.3 Esecuzione dell'esercizio	12
2.3.1 Timer	13
2.3.2 Cronometro	15
2.4 Memorizzazione dell'esercizio	17
2.5 Riproduzione di un esercizio in memoria	18
2.6 Posizionamento e dimensionamento dell'interfaccia	20
2.6.1 Esercizio in real time	21
2.6.2 Esercizio caricato da memoria	22
2.7 Guida rapida all'utilizzo	22
3 Sintesi del feedback audio	25
3.1 Tipi di feedback creati	26
3.2 Implementazione del feedback audio negli esercizi	27
3.2.1 Tipi di audio	27

3.2.2	L'audio negli esercizi	28
3.3	Comunicazione con il software dei suoni	31
3.3.1	Regole di creazione dei pacchetti : il protocollo OSC	31
3.3.2	Invio dei pacchetti: protocollo UDP	35
3.4	Software di generazione di suoni: Pure Data	37
4	Test sperimentali	41
4.1	Interfaccia in modalità test	41
4.2	Definizione dei task	43
4.3	Popolazione	45
4.4	Protocollo di esecuzione dei test	45
4.5	Memorizzazione dei task	48
5	Elaborazione dei dati e analisi statistica	51
5.1	Elaborazioni preliminari sui dati	51
5.1.1	Rimozione delle righe nulle	52
5.1.2	Calcolo della velocità della velocità del soggetto	52
5.1.3	Irregolarità del timer	57
5.2	Tipi di errore calcolati	59
5.2.1	Distanza media	59
5.2.2	Media del modulo della velocità relativa	60
5.2.3	Integrale del modulo della velocità relativa	62
5.2.4	Errore di posizione medio (o ritardo medio)	66
5.2.5	Lead error	69
5.2.6	Lag error	71
5.3	Analisi statistica	72
5.3.1	Selezione delle impostazione per l'analisi	72
5.3.2	Modalità di calcolo	73
5.3.3	Test non parametrici	73
5.4	Test di Wilcoxon con GraphPad Prism	75
5.4.1	Software di analisi statistica utilizzato	75
5.4.2	Test di Wilcoxon per dati appaiati	76

6 Interpretazione dei risultati	79
6.1 Distanza media	79
6.2 Media del modulo della velocità relativa	81
6.3 Integrale del modulo della velocità relativa	83
6.4 Errore di posizione medio	84
6.5 Lead error	87
6.6 Lag error	88
6.7 Riassunto dei risultati	90
Conclusioni	93
Bibliografia	95

Sommario

In seguito ad un ictus, frequentemente nei pazienti si denota una perdita della funzionalità motoria e della capacità di controllo del movimento stesso. Tale situazione, in sede di riabilitazione, viene evidenziata ad esempio con la misura di opportuni parametri cinematici nell'esecuzione di movimenti. Il feedback è un componente essenziale della riabilitazione, in particolare il feedback uditivo su parametri cinematici può risultare uno strumento molto utile per un migliore e più pronto recupero della capacità motoria.

Lo scopo di questo studio è quello di investigare gli effetti che diversi tipi di feedback audio portano nell'esecuzione di alcuni movimenti riabilitativi da parte dei pazienti.

E' stata presa in considerazione una popolazione di venti soggetti sani, privi di alcun tipo di menomazione motoria. Tre tipi di feedback audio sono stati sviluppati e forniti online durante l'esecuzione di sei esercizi di target tracking. Tutti i parametri cinematici di interesse dell'intero esercizio sono stati memorizzati per consentire la visualizzazione in momenti successivi dello stesso esercizio, e la valutazione dell'accuratezza dell'inseguimento attraverso il calcolo di svariati indici di errore. Sono stati comparati indici relativi ad esercizi di audio diverso ma eseguiti dallo stesso soggetto e la valutazione dell'effetto del feedback uditivo è stata ottenuta per mezzo di un test statistico sull'intero gruppo di soggetti.

Introduzione

Contesto: epidemiologia ictus

Secondo la definizione dell'OMS (Organizzazione Mondiale della Sanità) l'ictus è l'improvvisa comparsa di segni e/o sintomi riferibili a deficit focale e/o globale delle funzioni cerebrali, di durata superiore alle 24 ore. Questa patologia è una emergenza medica e deve essere prontamente diagnosticata e trattata in un ospedale per l'elevato rischio di disabilità e di morte che esso comporta. La definizione di ictus comprende, sulla base dei dati morfologici, l'ictus ischemico, più frequente, l'ictus emorragico, nel 15% dei casi e alcuni casi di emorragia meningea.

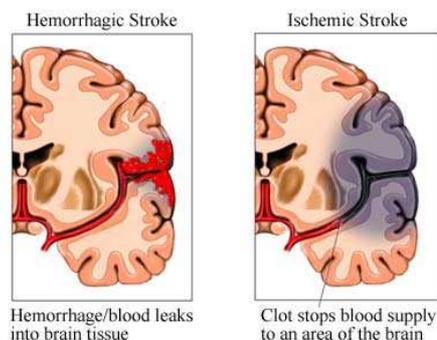


Figura 1: Rappresentazione grafica di un ictus emorragico e di un ictus ischemico

Nei Paesi industrializzati l'ictus rappresenta, per le sue dimensioni epidemiologiche e per il suo impatto socio-economico, una delle più importanti problematiche sanitarie, costituendo la prima causa di invalidità permanente e la seconda causa di demenza. I dati a riguardo sono a dir poco allarmanti: ogni anno si hanno 4.5 milioni di morti nel mondo di cui 400.000 solo in Europa a causa di questa grave patologia.

Nel nostro paese le malattie cerebrovascolari in generale hanno causato 246.415 ricoveri pari al 36.7% del totale dei ricoveri per malattie del sistema nervoso centrale e al 2.7% di tutti i ricoveri. Questa patologia può verificarsi ad ogni età, ma nel 50% dei casi i soggetti con ictus cerebrale hanno più di 70 anni, con una percentuale leggermente più alta negli uomini (7.4%) rispetto alle donne (5.9%)[1]. Il rischio di esserne colpiti aumenta con l'età: in pratica raddoppia ogni 10 anni a partire dai 45 anni, raggiungendo il valore massimo negli ultra-80enni [2]. Il progressivo invecchiamento della popolazione e la considerazione che l'incidenza dell'ictus è strettamente collegata all'età, fanno ritenere che in Italia l'incidenza di eventi ictali aumenterà nei prossimi anni: le proiezioni fino al 2016 fanno ritenere che tale aumento sarà circa del 22.2%.

L'aumento del tempo di sopravvivenza dopo l'ictus porterà a un aumento della prevalenza di tale patologia e della domanda di servizi sanitari per pazienti cronici stabilizzati con handicap neurologici. Il 40-50% dei soggetti che sopravvivono perde definitivamente la propria autonomia motoria mentre il 10% ha una recidività grave entro 12 mesi. In Italia oltre la metà dei sopravvissuti ha una disabilità nelle ADLs (Activities of Daily Living) . Secondo i dati di incidenza italiani si possono stimare circa 130.000 nuovi casi all'anno di ictus, circa la metà dei quali con disabilità a 6 mesi [3][4]. E' opinione ormai comune che i risultati della riabilitazione riguardo al danno motorio e funzionale sono strettamente legati al tempo che intercorre tra l'evento traumatico (l'ictus) e l'inizio della terapia.

Di fondamentale importanza risulta quindi un intervento specifico e immediato nella fase acuta, immediatamente successiva all'ictus. In passato l'assistenza al soggetto con ictus era affrontata con atteggiamento passivo.

Negli ultimi anni invece, dopo la pubblicazione e l'esperienza degli studi clinici che mostravano esiti migliori nei soggetti trattati in centri specifici, si sono diffuse diverse unità organizzate, le cosiddette Stroke Unit, per affrontare questa condizione. Pur nella variabilità degli assetti organizzativi e assistenziali sperimentali, le analisi effettuate hanno suggerito che possono essere considerati elementi indispensabili della Stroke Unit:

- La multidisciplinarietà di approccio da parte di un team esperto in malattie cerebrovascolari;

- La tempestività di avvio dell'intervento riabilitativo;
- L'applicazione di protocolli espliciti per la prevenzione delle complicanze maggiori;
- La messa in atto di interventi mirati alla formazione e all'aggiornamento continuo del personale;
- Il coinvolgimento di chi assiste il soggetto nel percorso riabilitativo;

Bisogna sottolineare però che in Italia queste strutture specializzate sono ancora molto poche. Se ne contano 89 in tutto il territorio nazionale di cui solo 8 nell'Italia meridionale. La realtà è che i pazienti sono soggetti solo ad una terapia hands-on, ovvero a diretto contatto con il fisioterapista. Purtroppo la qualità e la quantità delle terapie a cui si sottopone un soggetto si dimostra essere limitata dai costi che queste comportano e dalla disponibilità dei fisioterapisti nelle varie strutture ospedaliere. Risulta necessario, quindi, introdurre delle nuove terapie a basso costo che permettano al paziente di svolgere le sue attività riabilitative per periodi più lunghi e in modo continuato soprattutto nel periodo di fase acuta di questa patologia.

In questo background si inseriscono le numerose sperimentazioni dei due ultimi decenni. La ricerca ha portato a creare e ottimizzare delle nuove terapie riabilitative basate sull'uso di sistemi robotici. Queste tecniche innovative affiancate alla terapia tradizionale seguita da un fisioterapista possono migliorare le performance motorie del paziente e accorciarne i tempi di recupero. Un altro importante beneficio della riabilitazione tramite dispositivi robotici risiede nel fatto che si può quantificare, attraverso parametri analitici, se un paziente sta effettivamente rispondendo alla terapia a cui è sottoposto.

Dispositivi robotici per riabilitazione

Nell'ambito della riabilitazione dei pazienti affetti da ictus, lo sviluppo della ricerca si è indirizzato verso alcuni strumenti tecnologici innovativi. Tra le varie tecnologie la sperimentazione di sistemi robotizzati per il recupero della funzionalità motoria dell'arto superiore è di particolare interesse sia per la iniziale di-

mostrazione di efficacia clinica, evidenziata da preliminari pubblicazioni [5], sia per i futuri esiti che potrebbe avere in seguito allo sviluppo di nuove tecnologie.

I robot adibiti alla riabilitazione motoria si possono suddividere in due famiglie principali:

1. Esoscheletri
2. End effector base

I robot di prima generazione sono stati gli esoscheletri che però sono risultati assolutamente non modulabili, per le esigenze cliniche dei vari pazienti, molto costosi e particolarmente ingombranti e pesanti, per cui sono sostanzialmente stati abbandonati. Alcuni esempi di esoscheletri sono: il Pneu-WREX, in grado di esercitare un ampio range di forze sul braccio quando esso compie movimenti “naturali” e l’ARMin.



Figura 2: Immagine Pneu-WREX

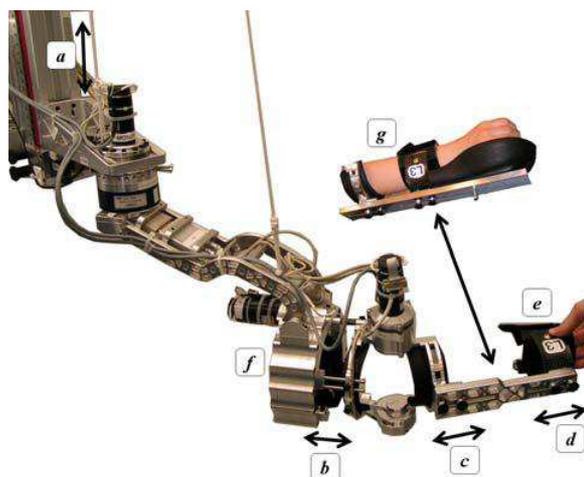


Figura 3: Rappresentazione dell’ARMin e dei diversi punti di regolazione (a, b, c, d, e, f, g)

La seconda generazione di robot ha visto la nascita di alcuni “bracci robotizzati” conosciuti anche come End effector base. Questi dispositivi permettono di interagire in modo del tutto sicuro fornendo al soggetto una sensazione di forza (feedback propriocettivo), una mobilizzazione passiva oppure assistenza alla mobilizzazione, una stimolazione neuro-sensoriale attraverso un sistema di feedback visivo ed uditivo basato sulla realtà virtuale. Questi sistemi sono in grado di misurare le caratteristiche cinematiche ed anche dinamiche del movimento dell’arto superiore con cui interagiscono.

Un esempio molto significativo di questa famiglia di dispositivi è il MIT-Manus .

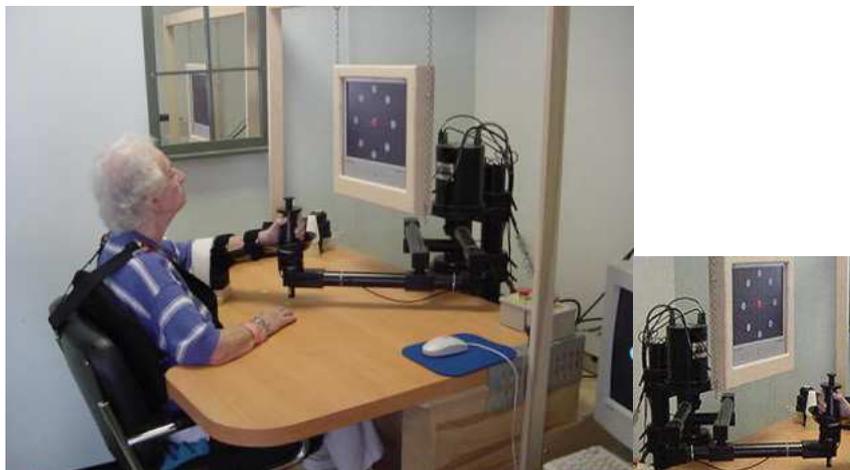


Figura 4: Esecuzione di un tipico esercizio riabilitativo con il MIT Manus

Questa macchina riconosce una componente volontaria del movimento, anche minimale, del paziente e quando questa componente intenzionale viene meno si attiva per aiutare il paziente stesso a terminare il movimento ed a raggiungere il bersaglio assegnato nell’esercizio. Questa soluzione permette di coniugare motricità volontaria residua del malato e raggiungimento del target con un movimento corretto, la combinazione di questi due fattori è fondamentale per riapprendere le sequenze motorie perse dopo l’ictus.

Il progetto di riabilitazione basato su questo dispositivo prevede 3 sedute settimanali di 40 minuti l’una per 6 settimane. I risultati ottenuti sono stati molto significativi dal punto di vista della rieducazione funzionale. Secondo gli esperti, i soggetti coinvolti nel progetto, hanno dimostrato una grandissima motivazione, anche in considerazione del fatto che, in circa 40 minuti di sessione, i pazienti

sono portati a fare un migliaio di movimenti. Altra particolarità tecnica è stata la progettazione dell'apparecchiatura secondo il criterio della “bassissima impedenza”, significa che il paziente non deve trovare nessuna resistenza da parte della macchina e che tutte le opposizioni meccaniche devono essere compensate dai motori per non compromettere la buona riuscita degli esercizi.

Nel settore della robotica riabilitativa anche l'Università di Padova ha dato il proprio contributo. Nei laboratori del DIMEG (Dipartimento di Innovazione Meccanica e Gestionale), infatti, è stato realizzato il Neuro-REhabilitation-roBOT (NeReBot), un interfaccia aptica a cavi, visibile in figura, che può essere utilizzata a letto dal paziente con l'obiettivo di riabilitare i soggetti post stroke in fase acuta.



Figura 5: Immagine del NeReBOT sviluppato nei laboratori del DIMEG

NeReBot è stato ideato e realizzato per effettuare diversi movimenti dell'arto superiore (spalla e gomito), simulando un trattamento hand-over-hand. La mobilizzazione passiva del NeReBot (sui 20 soggetti testati) sembra aver ridotto la debolezza e aver favorito il recupero della forza muscolare di pazienti post-stroke con arto superiore plegico/paretico [6].

Tutte queste nuove tecnologie di tipo meccanico, però, devono essere integrate alla motivazione e al coinvolgimento del paziente nella riabilitazione robot-assistita. Per ottenere questi effetti nel soggetto il metodo migliore e' quello di inserire nell'esecuzione dell'esercizio dei feedback audio o visivi. Secondo gli studi

di Reinkensmeyer, un dispositivo robotico integrato con un feedback interattivo può incrementare la motivazione del paziente nell'esecuzione del task motorio [7].

Per quanto riguarda il feedback audio esso verrà ampiamente descritto nel Capitolo 3 (vedi 3), mentre per quanto riguarda il feedback visivo, si può affermare che esso è identificato come un riscontro sul display di distruttori, parametri cinematici e punteggi ottenuti dal paziente durante la sessione di riabilitazione. Questo tipo di feedback ha un riscontro psicologico nel paziente stesso in quanto aumenta la sua attenzione sull'esercizio e lo invoglia a migliorare le proprie performance. Inoltre, la possibilità di fornire feedback della cinematica del movimento promuove l'apprendimento in soggetti sani e in pazienti colpiti da ictus.

Uno dei problemi principali di queste nuove metodiche resta la valutazione dell'efficacia dal punto di vista clinico e la quantificazione dei parametri che esse maggiormente influenzano. Date queste premesse, bisognerebbe istituire un adeguato programma che preveda task motori assistiti attraverso un interfaccia robotica nei pazienti paretici e con il feedback audio-visivo della performance motoria. Questa serie di sperimentazioni dovrebbe essere svolta su un più ampio ed omogeneo campione di pazienti post-ictus per verificarne la vera efficacia clinica.

Gli obiettivi principali che la robotica riabilitativa deve dunque porsi per diventare un adeguato e duraturo mezzo di riabilitazione sono:

1. Realizzazione di esoscheletri e sistemi aptici integrati a basso costo.
2. Creazione di task motori robot-assistiti nella prima fase post-ictus (fase acuta).
3. Valutazione dell'efficienza della terapia robot assistita in termini di riduzione delle ospedalizzazioni durante il progetto riabilitativo.
4. Valutazione del costo della industrializzazione del sistema robotico.

Capitolo 1

Feedback sonoro nella neuroriabilitazione

Alcune recenti ricerche in ambito neurologico hanno dimostrato che la stimolazione sonora è in grado di aumentare la plasticità celebrale, influenzando specifici meccanismi neuronali che possono contribuire al recupero da un danno neurologico, come può essere l'ictus.

In collegamento agli studi fatti da Sarkamo [8], la rielaborazione delle immagini che fotografano l'attività neuronale ha evidenziato che ascoltare musica produce degli indubbi effetti benefici sul cervello [9][?]. La musica, infatti, mette in contatto tra loro diversi parti del cervello, oltre alla corteccia uditiva, tra le quali: l'area frontale, l'area temporale, l'area parietale e l'area subcorticale. La connessione di tutte queste strutture provoca un aumento dell'attenzione, della memoria e delle funzioni motorie.

Suoni familiari possono facilitare e perfezionare particolari risposte motorie che prima erano associate a questi suoni [10]. Alcuni studi analoghi hanno dimostrato che un feedback sonoro ha prodotto netti miglioramenti in alcune ADLs (la distribuzione del peso nel sedersi, la camminata) in pazienti affetti da malattie neurologiche (Parkinson, SM, Ictus).

In seguito a queste ricerche in campo medico, l'ingegneria riabilitativa ha introdotto nei nuovi dispositivi medici un feedback sonoro che può agevolare il paziente sia nell'apprendimento dell'esercizio che a mantenere alta la propria

concentrazione su ciò che sta svolgendo.

L'auditory feedback, propriamente detto, denota quindi un segnale audio, generato automaticamente dal dispositivo, in risposta ad un'azione dell'utente o in risposta ad uno stato interno del sistema. I feedback possono essere suddivisi in due categorie [11][12]:

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

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

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

Un esempio molto significativo dell'utilizzo di questi feedback integrati in un robot-riabilitativo è dato dal Pneu-WREX. Al paziente, vincolato a questo esoscheletro, veniva chiesto di seguire un target su un display e, allo stesso tempo, di identificare un distruttore che compariva sullo schermo stesso. La presenza del distruttore diminuiva notevolmente le prestazioni del paziente, complicando l'inseguimento del target. L'introduzione di un feedback audio, il quale segnalava il superamento di un certo margine di errore, riportava le prestazioni a livelli soddisfacenti [13].

Come ho illustrato in tutti questi esempi, l'importanza di un efficace feedback audio acquista quindi ancora più valore negli esercizi di riabilitazione robot-assistiti. Il paziente in questa tipologia di riabilitazione, infatti, non ha l'ausilio del fisioterapista e il feedback audio è l'unico sistema che incoraggia il paziente stesso a mantenere la propria attenzione sull'esercizio e, quindi, a migliorarsi sempre.

Il lavoro di ricerca oggi si pone dunque come obiettivo quello di individuare quali siano i parametri più opportuni ed efficaci a cui legare un feedback

(acustico o visuale) in modo che esso possa essere veramente d'aiuto al paziente nell'esecuzione dell'esercizio.

1.1 Obiettivi

Il lavoro è principale di questo progetto è stato quello di implementare due diversi tipi di feedback audio (che dovranno essere integrati al Pneu-WREX) e il loro test su un gruppo di soggetti per verificare se essi agevolassero o meno l'esecuzione degli esercizi riabilitativi.

Prima di dedicarsi all'analisi e alla scelta di un adeguato feedback, si è dovuto realizzare un'interfaccia grafica per gestione le fasi creazione, visualizzazione, esecuzione e memorizzazione dei test.

Una volta realizzata tale interfaccia, si è posta l'attenzione allo studio di quale segnale informativo sonificare in forma di feedback.

Si è trattato quindi di trovare i giusti parametri su cui basare la funzione di feedback, come possono essere gli errori di velocità o di posizione tra paziente e target, e di capire in che modo deve essere realizzato il feedback al fine di essere il più utile possibile alla riabilitazione del paziente.

1.2 Software di elaborazione dei dati

Il primo obiettivo da raggiungere è stato quello di realizzare un'appropriata interfaccia grafica con il software Matlab.

Lo scopo era quello di ricreare un esercizio riabilitativo in cui il paziente dovesse inseguire un target su uno schermo, tramite l'utilizzo di una tavoletta grafica. Il target si muoveva lungo una traiettoria rettilinea piana che poteva essere, a seconda della modalità scelta, composta da tratti o a lunghezza costante o a lunghezza variabile.

Grazie alla funzione timer della libreria di Matlab è stato possibile a ricreare in tempo reale il movimento del target e del soggetto contemporaneamente.

Terminata la parte prettamente grafica, si è passati all'implementazione delle comunicazione fra applicazioni. In particolare, essa consisteva nella spedizione,

via UDP, al software Pure Data di un pacchetto dati contenente diversi valori tra i quali: le posizioni e le velocità lungo i 2 assi (x e y) di target, e soggetto. Da questi dati, sono state realizzate due tipologie diverse di feedback:

1. Legata alla velocità del target: molto simile ad un suono di rotolamento.
2. Legata all'errore di posizione del paziente rispetto al target.

Per questo primo studio sono state implementati solo questi due tipi di feedback. In eventuali lavori successivi sarebbe interessante poter ampliare la scelta fra i tipi di feedback disponibili, fornendo un ritorno di informazioni su parametri diversi da quelli presentati in questo studio, oppure ottimizzare i feedback qui presentati al fine di ottenere risultati ancor più attendibili.

1.3 Test su soggetti sani

Terminata la parte puramente informatica del lavoro, sono stati svolti i test su 20 soggetti per verificare se effettivamente l'ausilio del feedback permettesse loro di svolgere gli esercizi in modo migliore.

Ciascun soggetto eseguiva un test completo costituito da una serie di 6 task (Task A, Task B, Task C, Task D, Task E, Task F) diversi l'uno dall'altro, per tipo di traiettoria o tipo di feedback.

Per rendere questo test il più attendibile e omogeneo possibile è stato scelto di randomizzare l'ordine dei task per ciascun soggetto, in modo tale che ogni soggetto dovesse affrontare una serie di test diversa l'una dall'altra.

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

Capitolo 2

Software sviluppato

GUI è l'acronimo di graphical user interface (interfaccia grafica utente) e può essere definita, in un linguaggio non tecnico, come la “via” attraverso la quale un programma “dialoga” in maniera grafica con l'utente. In termini informatici si tratta di un *paradigma*, ossia un insieme di strumenti forniti da un linguaggio di programmazione per la stesura di programmi, che mira a consentire all'utente di interagire con il computer manipolando graficamente degli oggetti, svincolandolo dall'obbligo di imparare una serie di comandi da impartire da tastiera sulla riga di comando della console del sistema come invece avviene nelle tradizionali interfacce testuali CLI (command line interface). La GUI quindi permette all'utente di interagire agevolmente con la “macchina” attraverso una serie di dispositivi (mouse, tastiera, etc.) e simboli (icone, pulsanti, etc.) in maniera piacevole ed intuitiva.

2.1 Schema di funzionamento della GUI

Per agevolare tutte le operazioni durante l'esecuzione dei test sui soggetti è stata realizzata una GUI 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 l'applicazione esterna deputata alla generazione del feedback audio fornito durante gli esercizi di inseguimento

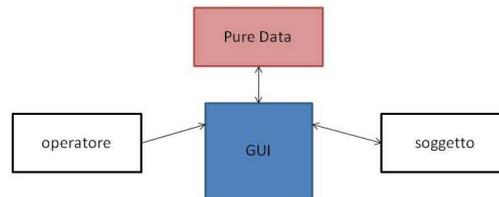


Figura 2.1: schema concettuale.

(Fig. 2.1).

In particolare l'interfaccia grafica (Fig.2.2) deve:

- consentire la creazione e la visualizzazione su grafico, 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 (Pure Data) deputata alla generazione del feedback audio

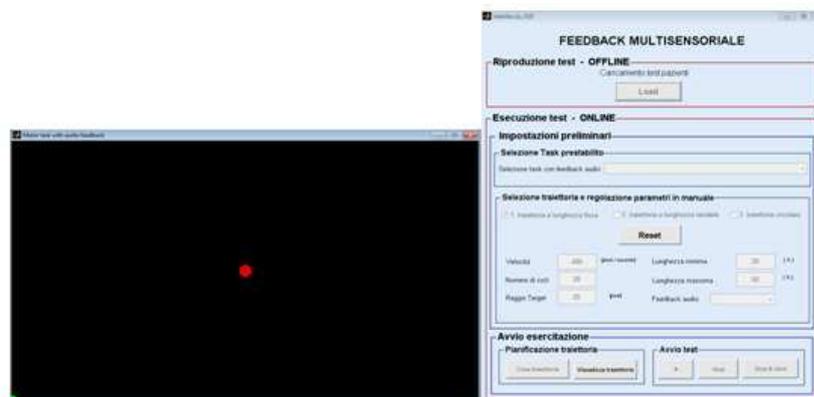


Figura 2.2: le due componenti dell'interfaccia grafica realizzata: pannello dei comandi per l'operatore, workspace per l'esecuzione dell'esercizio.

2.2 Creazione e visualizzazione dell'esercizio

L'interfaccia sviluppata permette la creazione di un esercizio riabilitativo consistente in un movimento destra-sinistra alternato nella sola direzione orizzontale. Di questo esercizio sono state previste due modalità distinte: uno primo tipo di esercizio in cui la lunghezza dei movimenti orizzontali si mantiene costante per tutta la durata, e un secondo esercizio in cui la lunghezza dei tratti di movimento varia in maniera casuale all'interno di un intervallo prestabilito. La sezione dell'interfaccia destinata alla creazione dell'esercizio (Fig. 2.3) prevede degli appositi controlli per stabilire nei minimi dettagli tutte le caratteristiche della traiettoria che si sta creando:

- velocità media di ogni tratto orizzontale di movimento (espressa in pixel al secondo)
- numero di tratti che compongono l'esercizio
- massima lunghezza possibile di un tratto orizzontale di movimento (espressa in percentuale sulla larghezza dello schermo su cui si vuole rappresentare l'esercizio)
- minima lunghezza possibile di un tratto orizzontale di movimento (espressa in percentuale sulla larghezza dello schermo su cui si vuole rappresentare l'esercizio)
- tipo di feedback audio associato all'esercizio

The screenshot shows a software interface titled "Selezione traiettoria e regolazione parametri in manuale". It features three radio buttons for trajectory selection: "1. traiettoria a lunghezza fissa" (unchecked), "2. traiettoria a lunghezza variabile" (checked), and "3. traiettoria circolare" (unchecked). Below these is a "Reset" button. The interface includes several input fields: "Velocità" (500 [pixel / secondo]), "Numero di cicli" (10), "Raggio Target" (25 [pixel]), "Lunghezza minima" (20 [%]), "Lunghezza massima" (90 [%]), and "Feedback audio" (a dropdown menu).

Figura 2.3: Sezione del pannello dei comandi dedicato alla creazione della traiettoria di esercizio

Scelte le impostazioni dei vari parametri, la creazione della traiettoria si articola in due fasi: la determinazione della coordinata (solo quella relativa alla direzione sulla quale si verifica il movimento, cioè l'orizzontale) di inizio e fine di ogni singolo tratto, e la pianificazione, secondo la legge desiderata, della traiettoria fra i punti iniziale e finale, precedentemente determinati, che identificano ogni singolo tratto. Va tenuto presente che la funzione che genera il vettore con gli estremi dei tratti della traiettoria richiede fra gli ingressi anche la massima lunghezza disponibile su cui calcolare quelle dei vari tratti che compongono l'esercizio. Questa lunghezza non corrisponde alla totale larghezza dello schermo su cui si rappresenta il movimento, bensì a questa viene tolta una quantità in pixel pari alla dimensione del raggio dell'indicatore del target in modo da evitare che, nel caso limite di una traiettoria con lunghezza massima pari al 100%, l'indicatore del target esca parzialmente dallo schermo nei punti di cambio di direzione. I punti di cambio direzione che tale funzione raccoglie in un vettore vengono forniti ad una seconda funzione che esegue il raccordo fra di essi attraverso una legge matematica, specificata come parametro di ingresso (legge a gradino, legge linear, legge ad accelerazione costante, legge a profilo trapezoidale di velocità, legge a polinomio cubico, legge a polinomio di quinto grado, legge sinusoidale, legge mista cubico-lineare).

Pianificazione di traiettorie per il moto punto-punto

Con il termine traiettoria si intende un percorso sul quale sia specificata una legge oraria di moto (velocità e accelerazione in ogni punto). Il problema della pianificazione di traiettoria è tipico del campo del controllo dei manipolatori e mira a stabilire le modalità con cui si vuole evolva il movimento di un manipolatore, da una postura iniziale a una postura finale. Solitamente il problema, a seconda del contesto, può essere risolto in due diverse modalità: pianificando traiettorie nello spazio operativo (si definisce il percorso dell'organo terminale del manipolatore nel comune spazio cartesiano) o pianificando traiettorie nello spazio di giunto (si specifica direttamente l'andamento desiderato per la posizione, la velocità e l'accelerazione dei singoli giunti). Tralasciando i problemi connessi al controllo del manipolatori, la stessa filosofia di pianificazione della traiettoria può essere

utilizzata per stabilire il moto del target, in termini di posizione, velocità a accelerazione, nel task riabilitativo. Si desidera infatti generare una funzione $q(t)$ che interpoli i valori assegnati per le variabili di posizione iniziale e finale di ogni tratto dell'esercizio, rispettando i vincoli imposti sui punti estremi e sul tempo di transizione. In particolare è stata presa in considerazione una *funzione polinomiale di quinto grado*:

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

sulla quale vengono specificate le seguenti condizioni al contorno:

- istante di tempo iniziale e finale
 - $t_i = 0$
 - $t_f = \frac{|q_f - q_i|}{v}$, con v velocità media del tratto
- posizione iniziale $q_i = q(t_i)$, e posizione finale $q_f = q(t_f)$
- velocità iniziale $\dot{q}_i = \dot{q}(t_i) = 0$, e velocità finale $\dot{q}_f = \dot{q}(t_f) = 0$
- accelerazione iniziale $\ddot{q}_i = \ddot{q}(t_i) = 0$, e accelerazione finale $\ddot{q}_f = \ddot{q}(t_f) = 0$

imposte sul polinomio:

$$q(t) = a_0 + a_1(t - t_i) + a_2(t - t_i)^2 + a_3(t - t_i)^3 + a_4(t - t_i)^4 + a_5(t - t_i)^5$$

Si ottengono quindi i valori dei coefficienti che identificano univocamente la traiettoria cercata:

$$\begin{cases} a_0 = q_i \\ a_1 = \dot{q}_1 \\ a_2 = \frac{1}{2}\ddot{q}_1 \\ a_3 = \frac{20(q_f - q_i) - (8\dot{q}_f + 12\dot{q}_i)T + (3\ddot{q}_f - 2\ddot{q}_i)T^2}{2T^3} \\ a_4 = \frac{30(q_i - q_f) + (14\dot{q}_f + 16\dot{q}_i)T + (3\ddot{q}_f - 2\ddot{q}_i)T^2}{2T^4} \\ a_5 = \frac{12(q_f - q_i) - 6(\dot{q}_f + \dot{q}_i)T - (\ddot{q}_f - \ddot{q}_i)T^2}{2T^5} \\ T = t_f - t_i \end{cases}$$

La traiettoria creata non presenta discontinuità in nessuno dei profili di interesse: posizione, velocità e accelerazione.

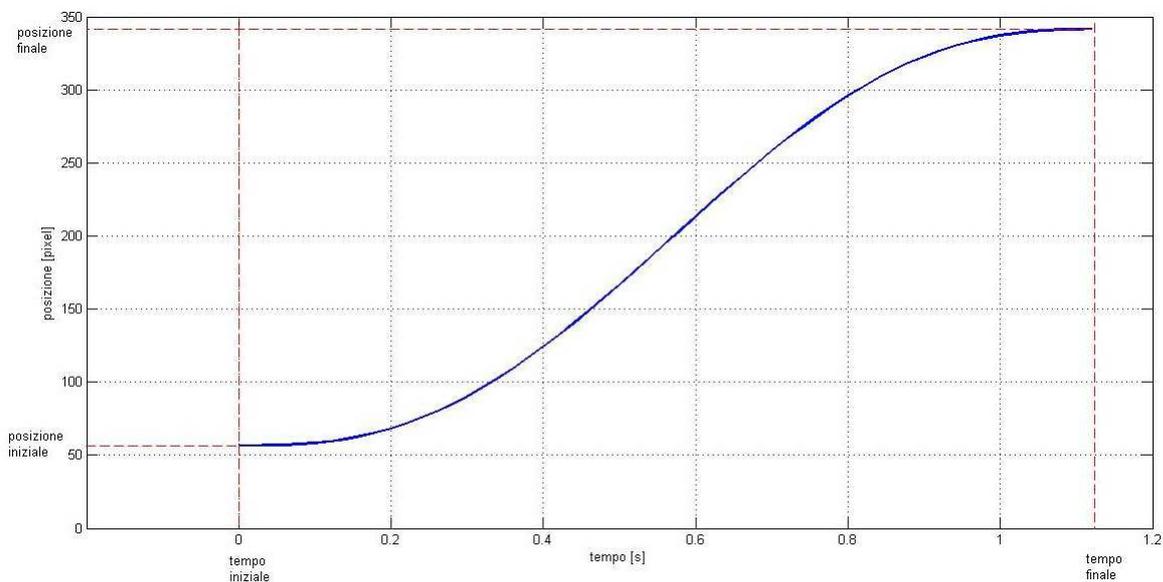


Figura 2.4: grafico di posizione per un generico tratto del task ottenuto attraverso pianificazione con polinomio di quinto grado

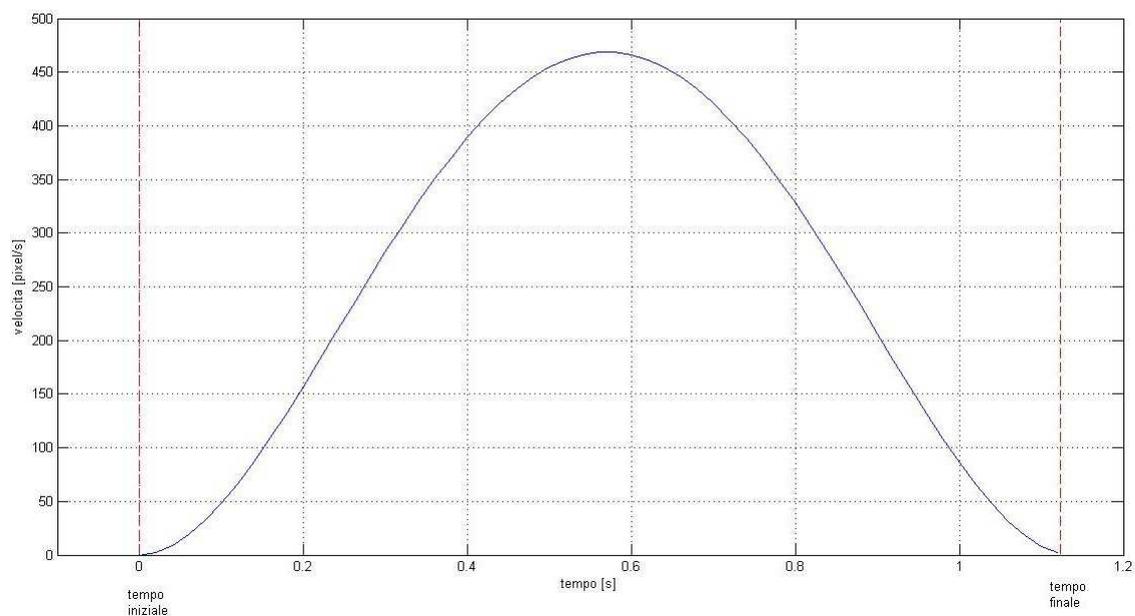


Figura 2.5: grafico di velocità per un generico tratto del task ottenuto attraverso pianificazione con polinomio di quinto grado

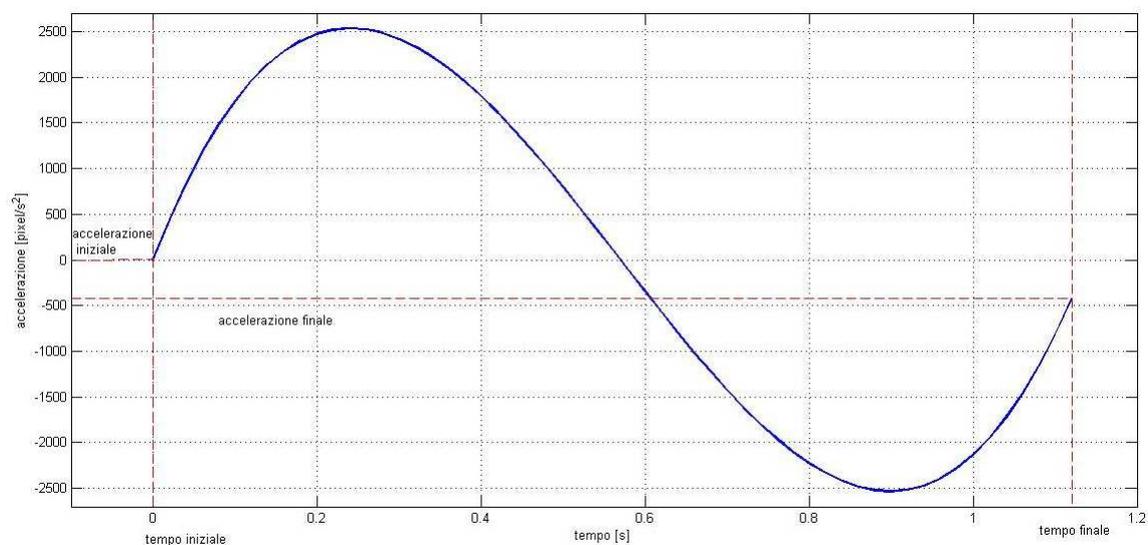


Figura 2.6: grafici dell'accelerazione di un generico tratto di un task ottenuti con polinomio di quinto grado. Si noti l'imprecisione della pianificazione adottata nel profilo di accelerazione che provoca un'accelerazione finale del tratto non nulla.

Per quanto riguarda la visualizzazione della traiettoria creata, essa viene rappresentata in una figura a sè stante con un grafico, opportunamente scalato sulla dimensione dello schermo su cui avverrà l'esecuzione dell'esercizio, che riporta anche i limiti della traiettoria prescelta e quelli tipici dello schermo (Fig. 2.7)

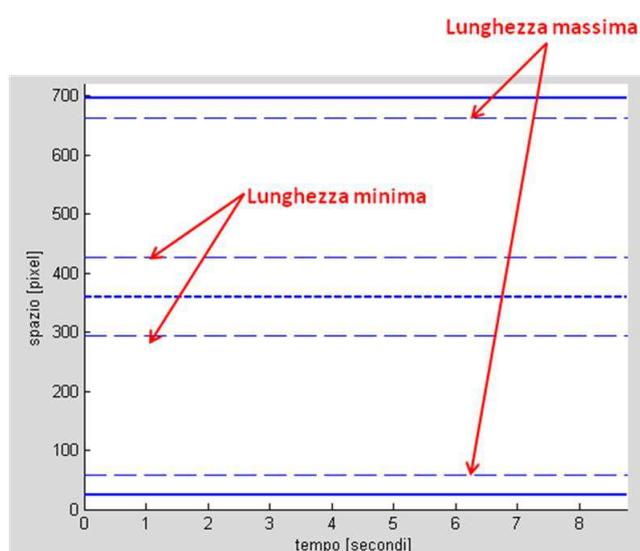


Figura 2.7: Visualizzazione dei limiti della traiettoria.

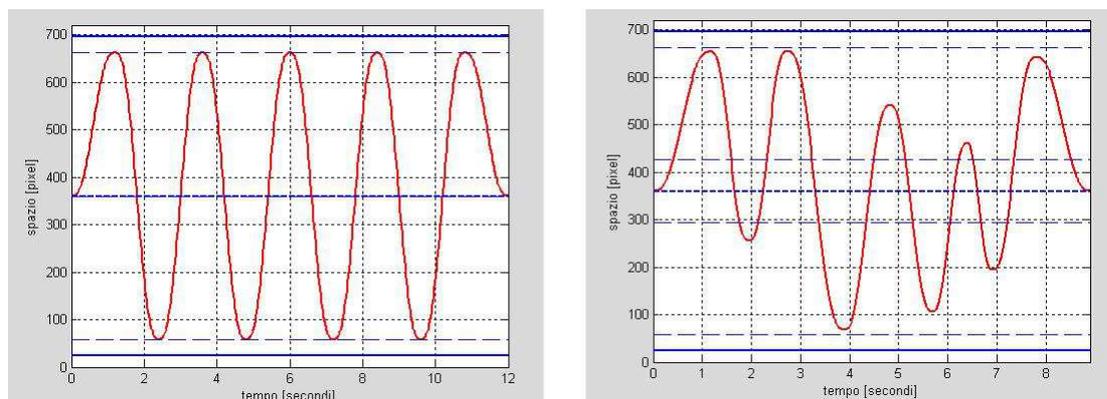


Figura 2.8: Esempi di visualizzazione di due traiettorie.

2.3 Esecuzione dell'esercizio

La riproduzione dell'esercizio avviene in una finestra dedicata dove sono visibili gli indicatori del soggetto e del target, cioè degli oggetti grafici di forma circolare con dimensione impostabile e di diverso colore, verde e rosso rispettivamente. Nello specifico gli indicatori del soggetto e del target sono degli oggetti grafici di tipo *hgtransform* per i quali si possono impostare le proprietà:

Parent handle della figura associata all'oggetto (cerchio con riempimento colorato)

Matrix matrice di trasformazione 4x4 applicata all'oggetto che consente di modificare la sua posizione e il suo orientamento

Per quanto riguarda posizionamento e dimensionamento della finestra nella quale avviene l'esecuzione dell'esercizio, essi dipendono dal contesto in cui si sta lavorando (modalità on line, modalità off line, utilizzo del doppio schermo, ecc.).

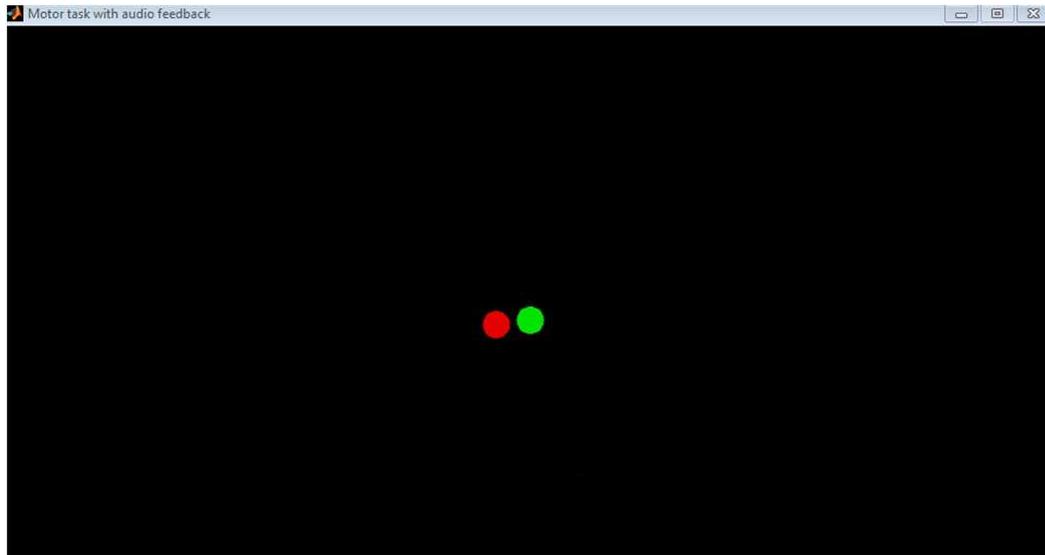


Figura 2.9: finestra di esecuzione dell'esercizio con target (rosso) e soggetto (verde).

L'idea che sta alla base della riproduzione a video dei movimenti di paziente e target consiste nel posizionare, con la corretta temporizzazione, gli indicatori di target e paziente a seconda dei valori nel vettore della traiettoria per quanto riguarda il target, e di quelli che si acquisiscono attraverso la funzione $get(\theta, 'PointerLocation')$ che restituisce le coordinate sullo schermo della periferica manipolata dal soggetto. Tale risultato si ottiene attraverso l'utilizzo combinato di due oggetti forniti da MatLab: il timer ed il cronometro.

2.3.1 Timer

L'oggetto timer fornito da MatLab consente di eseguire una determinata sequenza di istruzioni, o una intera funzione, in maniera ripetitiva e sincronizzata nel tempo, specificando la frequenza con la quale si voglia che ciò avvenga. Le proprietà fondamentali dell'oggetto che vanno impostate per un suo corretto funzionamento sono :

TimerFcn funzione, o sequenza di istruzioni, che viene eseguita ad ogni periodo del timer

Period periodo di tempo che intercorre tra due successive chiamate della funzione `TimerFcn`

ExecutionMode modalità con cui viene valutato l'intervallo di tempo tra due task successivi

Per quanto riguarda quest'ultima proprietà si può scegliere fra tre diverse modalità di esecuzione: “`fixedRate`”, “`fixedDelay`”, “`fixedSpacing`” (Fig. 2.10). La modalità “`fixedRate`” è quella che più si mantiene vicina alla sincronizzazione impostata attraverso la proprietà “`Period`”, in quanto si inizia a conteggiare il periodo di tempo fra due esecuzioni successive della funzione del timer immediatamente dopo che la chiamata alla funzione è stata aggiunta coda di esecuzione di MatLab.

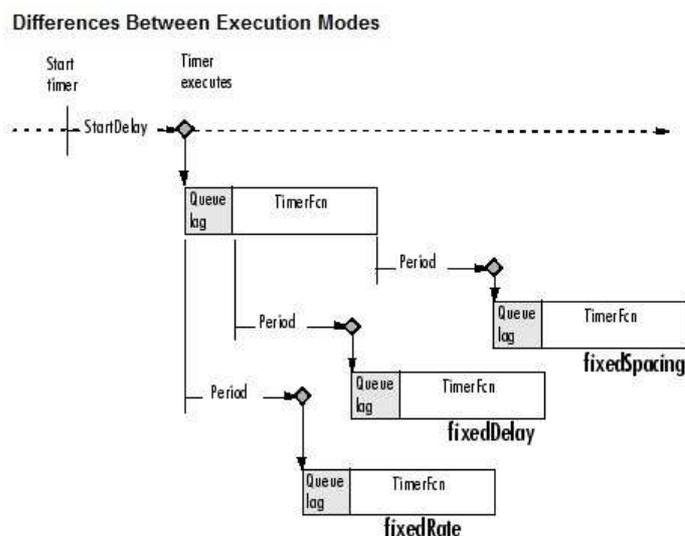


Figura 2.10: differenze fra le modalità di esecuzione del timer.

Con riferimento alla figura, si noti che l'area ombreggiata (queue lag) rappresenta un intervallo di tempo indeterminato che intercorre fra l'istante in cui la chiamata alla funzione del timer viene messa nella queue di esecuzione e quello in cui la funzione viene realmente eseguita. La durata di questo ritardo dipende da quali processi MatLab sono attivi al momento della chiamata.

Per visualizzare il progresso di un esercizio in real time è sufficiente avviare il timer e affidare alla sua funzione `TimerFcn` il compito di graficare a video gli indicatori di target e soggetto. Alla funzione del timer pertanto vengono forn-

ti come parametri, per quanto riguarda la sola visualizzazione dell'esercizio, i vettori delle traiettorie del target (corredati di relativo vettore dei tempi) e gli oggetti hgtransform di target e soggetto. Ad ogni chiamata si andrà a modificare la matrice di trasformazione degli oggetti indicatori con la riga opportuna della traiettoria, per quanto riguarda il target, e con la posizione del puntatore aggiornata, per quanto riguarda il soggetto, ottenendo così il movimento degli indicatori sullo schermo. La sensazione di fluidità del movimento può essere migliorata aumentando la frequenza di esecuzione del timer, cioè modificando la proprietà "Period".

2.3.2 Cronometro

L'utilizzo del cronometro si rende necessario per mantenere costante la frequenza di rappresentazione dell'esercizio. Infatti nonostante si sia scelta come modalità di esecuzione la tipologia "fixedRate", che dovrebbe mantenere le imposizioni temporali fra le chiamate indipendentemente dal tipo e dalla complessità delle operazioni che vengono eseguite, si nota che il timer tende nel proseguo dell'esercizio a ritardare, portando a compimento l'esecuzione della traiettoria in un tempo più lungo di quello pianificato. Per garantire la coerenza fra tempi di pianificazione e tempi di esecuzione dell'esercizio si fa ricorso all'oggetto cronometro che MatLab mette a disposizione. Esso viene avviato in corrispondenza del timer e ad ogni chiamata della TimerFcn si va a valutare l'intertempo, cioè il *reale* intervallo di tempo trascorso dall'avvio dell'esercizio. L'intertempo viene utilizzato per "pescare" nel vettore della traiettoria, la posizione del campione relativo all'istante di tempo di pianificazione che corrisponde a quello reale di esecuzione dell'esercizio fornito appunto dall'intertempo. In questo modo non si andranno a rappresentare tutti i campioni della traiettoria pianificata, tuttavia vi sarà la corrispondenza fra ciò che si è progettato e ciò che si va a visualizzare come esercizio, condizione indispensabile per garantire una corretta esecuzione dei test, con dei dati che siano consistenti per la successiva analisi (Fig. 2.11).

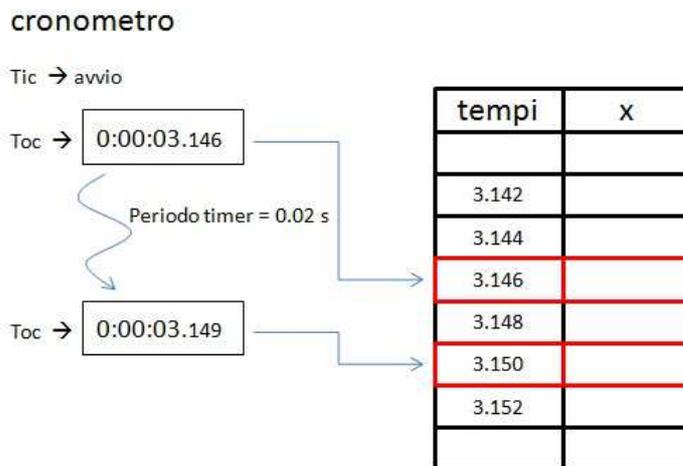


Figura 2.11: cronometro per il plotaggio degli indicatori

L'utilizzo del cronometro di MatLab è molto semplice e avviene per mezzo dei due soli comandi:

- tic: azzeramento e avvio del cronometro
- toc: intertempo

La sezione del pannello dei comandi dedicata al controllo dell'esecuzione (Fig. 2.12) permette l'avvio, la pausa e l'interruzione dell'esercizio. Queste due ultime scelte impongono l'arresto del timer, inoltre la modalità pausa richiede una particolare gestione del cronometro per memorizzare l'istante temporale in cui è avvenuta la sospensione dell'esercizio in modo che, i nuovi intertempi calcolati al riavvio, tengano conto dell'offset temporale. Ciò viene fatto memorizzando all'atto della pausa l'indice della posizione corrispondente al momento di pausa nel vettore dei tempi della traiettoria.



Figura 2.12: Comandi per il controllo dell'esecuzione.

2.4 Memorizzazione dell'esercizio

La possibilità di effettuare un'analisi sulle prestazioni dei soggetti nell'esecuzione dei test è strettamente correlata alla possibilità di avere a disposizione i tracciati dell'inseguimento e tutti i dati relativi all'esercizio eseguito. E' necessario pertanto implementare nell'interfaccia un controllo per memorizzare l'esecuzione di tutti i task che vengono eseguiti. Ad ogni creazione di un task viene predisposta una struttura dati (vedi 4.5) in grado di memorizzare tutte le informazioni utili per la riproduzione dell'esercizio considerato in un secondo momento.

Per quanto riguarda la memorizzazione dello svolgimento dell'esercizio di particolare interesse è il campo della struttura che contiene la matrice dei tracciati di target e soggetto. Nelle colonne della matrice vengono memorizzate le seguenti informazioni:

1. tempo
2. coordinata x della posizione del soggetto
3. coordinata y della posizione del soggetto
4. coordinata x della posizione del target
5. coordinata y della posizione del target
6. velocità del target nella direzione x
7. velocità del target nella direzione y
8. velocità del soggetto nella direzione x
9. velocità del soggetto nella direzione y
10. numero di sospensioni-riavvi dell'esercizio
11. indicatore del tipo di audio associato all'esercizio
12. numero del tratto

Si noti che la colonna relativa al tempo non memorizza gli istanti temporali della pianificazione, bensì quelli in cui avviene l'effettiva rappresentazione degli oggetti indicatori, cioè i valori degli intertempi restituiti dal cronometro.

Le righe della matrice vengono riempite progressivamente all'esecuzione dell'esercizio, una per ogni chiamata della funzione associata al timer, in modo che registrino fedelmente ciò che accade durante l'esecuzione. Una volta terminato l'esercizio la relativa struttura dati contenente la matrice opportunamente completata può essere memorizzata in un file dati “*.m” (Fig. 2.13).

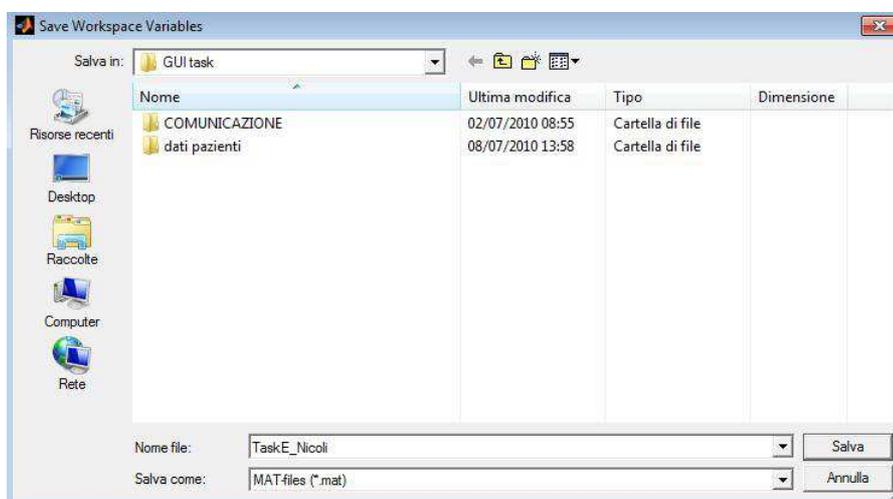


Figura 2.13: finestra di dialogo per la memorizzazione di un esercizio.

2.5 Riproduzione di un esercizio in memoria

La memorizzazione dell'esercizio, oltre ad essere indispensabile per consentire l'analisi statistica dei task, permette di realizzare un'altra funzionalità con cui arricchire l'interfaccia progettata, cioè quella di riprodurre l'esercizio così come era stato eseguito originariamente. Ciò può risultare utile per una prima analisi qualitativa sulle performance dei vari soggetti. Inoltre la possibilità di rivedere l'esercizio è uno strumento ausiliario comodo per trarre considerazioni complementari a quelle fornite dall'analisi statistica e per giustificare risultati non previsti dal punto di vista numerico, dovuti a situazioni singolari verificatesi durante l'esercizio (errori di inseguimento dovuti a disattenzioni momentanee, correzioni

troppo brusche, ecc.) rilevabili solo tramite la visione delle registrazioni. Un'aspetto che rende ancor più interessante questa funzione è la possibilità di risentire pure il feedback audio fornito durante l'esecuzione, consentendo così all'operatore di effettuare un controllo immediato fra feedback audio generato e risposta del soggetto allo stesso. Per caricare un'esercizio presente in memoria ci si serve dell'apposita finestra di dialogo (Fig.2.14).

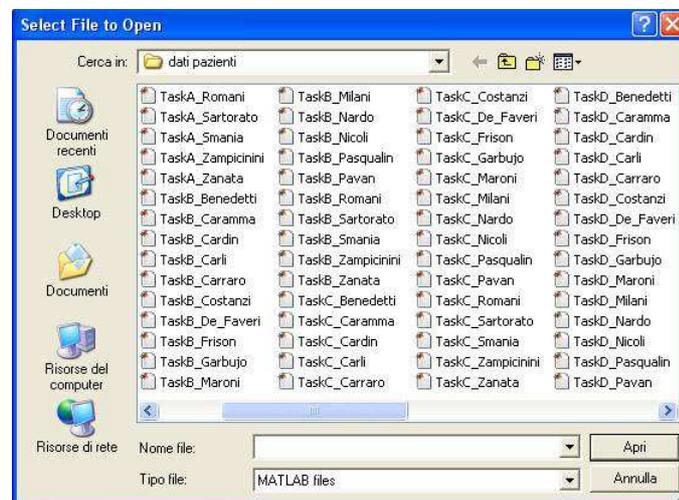


Figura 2.14: finestra di dialogo per il caricamento di un esercizio.

La visualizzazione di un esercizio caricato da memoria è affidata sempre alla funzione associata all'oggetto timer, come avveniva per l'esercizio in real time, con la differenza che ora tutti i dati relativi alle posizioni che devono assumere gli indicatori del target e del soggetto sono già pronti, e che non è necessario creare e riempire la struttura dati per la memorizzazione dell'esercizio dal momento che si tratta di un esercizio già registrato.

Il diverso comportamento che deve assumere la funzione specificata dalla proprietà `TimerFcn` è stabilito dal valore di una variabile flag impostata ad 1 quando si va a caricare un esercizio memorizzato attraverso l'apposita finestra di dialogo o a 0 quando si va a creare una nuova traiettoria di esercizio.

2.6 Posizionamento e dimensionamento dell'interfaccia

Come già accenato in precedenza l'interfaccia si compone di due finestre (Fig. 2.2):

- finestra principale con il pannello dei comandi per la gestione delle esercitazioni (rivolta all'operatore)
- finestra di rappresentazione su cui appare l'esercizio in corso (rivolta al soggetto)

La finestra dei comandi viene sempre posizionata nella parte destra dello schermo principale (quello relativo alla postazione di lavoro dell'operatore). La finestra di esercizio invece può essere posizionata e dimensionata variatamente a seconda del contesto di lavoro. Si tratta di una finestra nella quale la proprietà "CloseRequestFcn" viene abilitata solamente quando si va a chiudere la finestra dei comandi e alla quale sono state rimosse la barra degli strumenti e la barra dei menù. È composta da un piano cartesiano dallo sfondo completamente nero i cui assi, resi invisibili assieme alla rispettiva griglia, sono posizionati nell'estremo in basso a sinistra dello schermo. All'interno di questo spazio ("workspace" dell'esercizio) verrà visualizzato il movimento del target e del soggetto (Fig. 2.15).

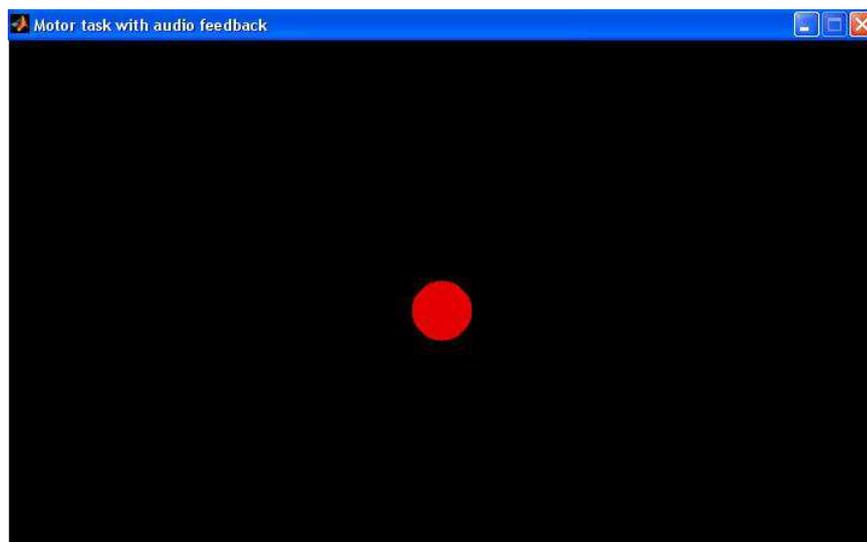


Figura 2.15: finestra di esercizio.

2.6.1 Esercizio in real time

La configurazione ideale è quella di disporre di due schermi separati, uno per la finestra dei comandi, l'altro per la finestra di esercizio. Per il corretto posizionamento si utilizza l'istruzione `get(0, 'MonitorPosition')` che restituisce delle righe contenenti le informazioni relative a tutti gli schermi identificati. In particolare ogni riga riporta:

- coordinata x di origine dello schermo (punto in basso a sinistra)
- coordinata y di origine dello schermo (punto in basso a destra)
- larghezza (in pixel)
- altezza (in pixel)

La finestra dei comandi viene sempre posizionata nello schermo a sinistra in modo che seguendo la direzione crescente delle coordinate in x si passi dallo schermo dell'operatore a quello dell'esercizio.

Bisogna porre attenzione al fatto che i movimenti di target e soggetto sono sempre definiti in termini di coordinate relative alla finestra di esercizio, ciò comporta la sottrazione dell'offset di posizionamento in x della finestra dove necessario, come ad esempio nell'acquisizione della posizione del soggetto.

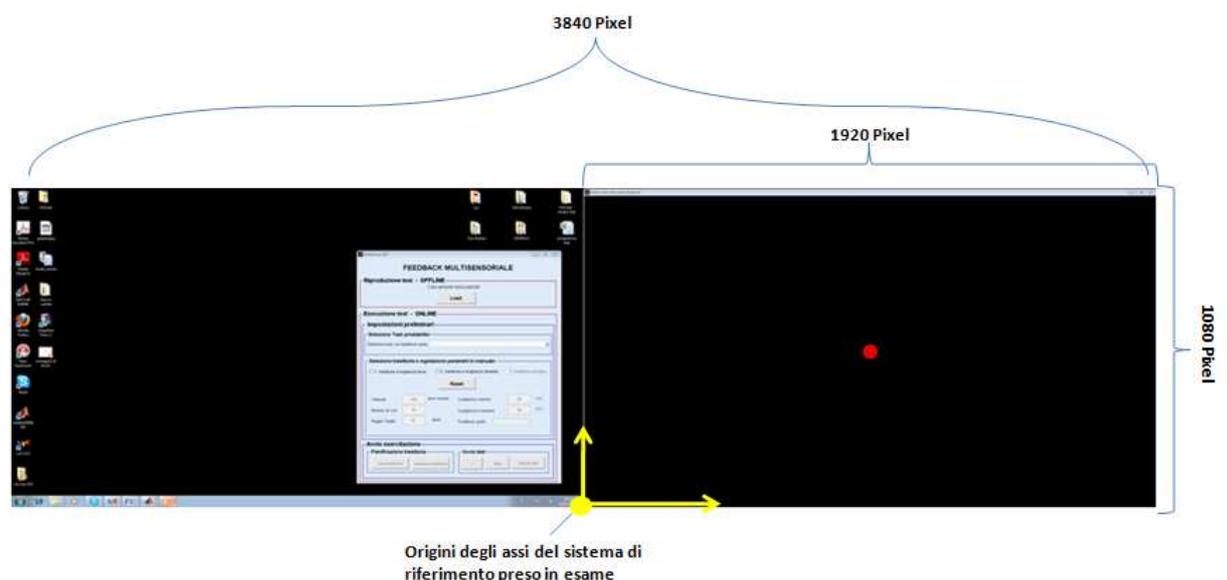


Figura 2.16: Esempio di posizionamento delle finestre dell'interfaccia.

Quando si ha a disposizione un solo schermo (situazione non comunque non compatibile per l'esecuzione di test, ma solo per verifiche di funzionamento), la finestra di esercizio viene visualizzata sullo schermo principale, alla sinistra di quella dei comandi, prevedendo un fattore di scala pari a 0.5 per consentire la visualizzazione affiancata delle due finestre.

2.6.2 Esercizio caricato da memoria

Quando l'esercizio è off line, cioè viene caricato da memoria, la situazione ideale è quella di poter agevolmente tenere sotto controllo entrambe le finestre per avviare e arrestare a piacere l'esercizio e contemporaneamente osservarne l'evoluzione sull'apposita finestra. La soluzione che meglio si adatta a questa necessità è quella di visualizzare entrambe le finestre sullo stesso schermo, lasciando a destra quella dei comandi e sistemando nella parte di sinistra, dopo un opportuna scalatura con un fattore 0.5, la finestra di esercizio.

Va ricordato che ogni qualvolta si esegue una scalatura essa interessa tutte le dimensioni in gioco, dalle dimensioni dello schermo, a quelle degli indicatori, fino alle traiettorie, in modo che siano mantenute sempre le stesse proporzioni fra le grandezze degli oggetti che interessano lo svolgimento (o la sola visione) dell'esercizio.

2.7 Guida rapida all'utilizzo

Di seguito viene riportata una breve e sommaria guida per l'utilizzo di tutte le funzionalità implementate nell'interfaccia (Fig.2.17).

- **tasto Load:** apre la finestra di dialogo per il caricamento di un esercizio presente in memoria
- **selezione task prestabilito:** menù "a tendina" che permette di scegliere fra una serie di esercizi con impostazioni e traiettoria già definite, utilizzati nella sessione di test sui soggetti

- selezione tipo traiettoria: serie di check box per la scelta del tipo di traiettoria che si vuole realizzare
- velocità: velocità media di ogni tratto della traiettoria, espressa in pixel al secondo
- numero cicli: numero di tratti rettilinei che compongono la traiettoria
- raggio target: dimensioni in pixel del raggio degli oggetti circolari che rappresentano il target e il soggetto
- lunghezza minima: percentuale della dimensione dello schermo che rappresenta la minima lunghezza possibile per un singolo tratto della traiettoria (controllo valido solo per le traiettorie con tratti a lunghezza variabile)
- lunghezza massima: percentuale della dimensione dello schermo che rappresenta la massima lunghezza possibile per un singolo tratto della traiettoria (nel caso di traiettorie a lunghezza dei tratti fissa il valore calcolato a partire da questa percentuale rappresenta la lunghezza di ogni tratto)
- feedback audio: menù “a tendina” che permette la scelta del tipo di audio da associare alla traiettoria durante l’esecuzione dell’esercizio
- tasto reset: reimposta tutti i parametri ai valori di default e cancella, se presente, la traiettoria creata ripristinando la situazione iniziale
- tasto crea traiettoria: crea la traiettoria a partire dalle impostazioni specificate nei vari campi dell’interfaccia
- tasto visualizza traiettoria: apre la finestra per la visualizzazione del grafico della traiettoria creata\caricata
- tasto >: avvia (o riprende) l’esecuzione dell’esercizio
- tasto ||: sospende l’esecuzione dell’esercizio. Questo tasto non rappresenta una scelta possibile quando si sta eseguendo uno dei task già predisposti poichè si suppone che l’utilizzo di tali task avvenga in sede di test, dove non è consentito interferire con l’azione del soggetto.

- tasto stop: blocca l'esecuzione dell'esercizio
- tasto stop&save: blocca l'esecuzione dell'esercizio e apre la finestra di dialogo per la memorizzazione della struttura dati contenente tutte le informazioni relative al task appena eseguito.

FEEDBACK MULTISENSORIALE

Riproduzione test - OFFLINE
Caricamento test pazienti
Load

Esecuzione test - ONLINE

Impostazioni preliminari

Selezione Task prestabilito
Selezione task con feedback audio [dropdown]

Selezione traiettoria e regolazione parametri in manuale

1. traiettoria a lunghezza fissa 2. traiettoria a lunghezza variabile 3. traiettoria circolare

Reset

Velocità: 500 [pixel / secondo] Lunghezza minima: 20 [%]
Numero di cicli: 10 Lunghezza massima: 90 [%]
Raggio Target: 25 [pixel] Feedback audio: [dropdown]

Avvio esercitazione

Pianificazione traiettoria
Crea traiettoria Visualizza traiettoria

Avvio test
> Stop Stop & Save

Figura 2.17: pannello dei comandi per la gestione degli esercizi

Capitolo 3

Sintesi del feedback audio

Lo scopo di questo studio è quello di investigare se l'aggiunta di un feedback audio può modificare la qualità della traiettoria di inseguimento del target da parte del soggetto in esercizi riabilitativi. Nell'ambito del target tracking il feedback visivo rappresenta il canale dominante dal quale il soggetto riceve informazioni riguardo la precisione del suo inseguimento, tuttavia il feedback audio può integrare questa modalità di ritorno di informazioni andando ad amplificare ed enfatizzare piccoli errori di posizione, non visibili a causa della limitata risoluzione del feedback video. Inoltre il suono ben si adatta a fornire informazioni legate alla velocità dei movimenti, le quali sono di difficile derivazione dal feedback visivo e richiedono delle complesse elaborazioni da parte del soggetto.

Si tratta quindi di verificare l'ipotesi secondo la quale un feedback audio che fornisce informazioni sugli errori di movimento e su come correggerli possa facilitare il raggiungimento del movimento richiesto, incrementando il livello di attenzione o la velocità di apprendimento del soggetto, e alimentare la motivazione durante l'esercitazione.

L'interfaccia implementata, ed in particolare il gruppo di esercizi che essa permette di creare, rappresenta lo strumento attraverso il quale testare l'efficacia e l'utilità del feedback uditivo. A questo scopo sono state create due diverse modalità di feedback audio, fornite on line al soggetto durante l'esecuzione dei test.

3.1 Tipi di feedback creati

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

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

Sulla base di queste considerazioni sono state create quattro diverse tipologie di feedback audio:

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

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

Gli altri due tipi di audio implementati realizzano una sintesi vocale. La modulazione del suono è legata all'errore di posizione su entrambe le dimensioni del piano, in particolare:

- l'errore di posizione in x definito come $|x_t - x_s|$ viene mappato sul volume e sulla frequenza del suono
- l'errore di posizione in y definito come $|y_t - y_s|$ viene mappato sulla formante, ovvero la coppia di frequenze che produce una vocale

Spazializzazione del suono

Entrambe le tipologie di audio sono disponibili in due modalità: spazializzata e non spazializzata.

Per spazializzazione si intende un insieme di tecniche per l'elaborazione dei segnali sonori attraverso le quali una sorgente virtuale di suono può essere posizionata in un qualunque punto dello spazio attorno all'ascoltatore.

La spazializzazione delle informazioni audio incrementa il realismo dell'ambiente virtuale di lavoro, portando alla creazione di un vero e proprio un paesaggio sonoro (soundscape) tridimensionale, all'interno del quale le sorgenti assumeranno una posizioni diverse rispetto all'ascoltatore e potranno anche "muoversi" intorno a lui 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.

3.2 Implementazione del feedback audio negli esercizi

3.2.1 Tipi di audio

Le tipologie di suono implementate nell'interfaccia si possono distinguere in due categorie:

- feedback uditivi : successioni di suoni forniti durante l'esecuzione dell'esercizio, con funzione di componente informativa aggiuntiva per il completamento del task
- segnali sonori di inizio e fine esercizio : semplici segnali acustici per richiamare l'attenzione del paziente riguardo l'imminente avvio dell'esercitazione e per informarlo sul completamento dello stesso

I primi sono già stati descritti in precedenza. I secondi non sono altro che dei semplici 'beep' forniti in forma di countdown o di unico impulso ("single shot") per segnalare rispettivamente l'inizio e la fine dell'esercizio. Si tratta essenzialmente di un accorgimento per migliorare la comodità nell'esecuzione dell'esercizio, tuttavia non va sottovalutato il fatto che questi semplici segnali possono risultare utili per il soggetto e meglio predisporlo in termini di attenzione e confortevolezza all'approccio al test.

3.2.2 L'audio negli esercizi

Il fatto che il feedback audio, per poter avere effetti sull'inseguimento del target da parte del soggetto, vada fornito on line all'esecuzione dell'esercizio, suggerisce che le modifiche implementative per l'utilizzo del suono vadano inserite nella parte in cui si programma la visualizzazione a video dell'esercitazione, cioè nella funzione associata alla proprietà `TimerFcn` dell'oggetto `timer`.

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. Ad ogni segnale è stato univocamente associato un valore intero (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 (Tab.3.1).

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

Tabella 3.1: tabella riassuntiva dei flag utilizzati per i diversi tipi di audio.

Esercizio on line

L'esecuzione di un esercizio in modalità on line comporta il coinvolgimento di entrambi i tipi di segnali sonori: feedback e segnali di inizio e fine esercizio. Infatti l'avvio di ogni esercizio è preceduto da una fase preliminare nella quale viene riprodotto un sorta di conto alla rovescia di tre secondi, all'esaurimento del quale il target inizia il suo movimento. In questa fase la funzione del timer, ad ogni sua chiamata, non esegue nessuna operazione se non quella di inviare all'applicazione che genera il suono gli appositi pacchetti di comunicazione dove il corrispondente campo per il tipo di audio sia composto da una successione correttamente temporizzata di 0 e 9, con cambi ad ogni 0.5 secondi. Si ottiene così la simulazione di un countdown costituito da una successione di "beep" della durata di 0.5 secondi intervallati da altrettante pause della stessa durata.

Al termine del countdown, la funzione del timer inizia a processare il vettore della traiettoria e ad acquisire le coordinate del soggetto andando a rappresentare a video lo svolgimento dell'esercizio. In questa fase il flag contenuto nei pacchetti per l'applicazione esterna contiene il valore relativo al feedback audio che si è impostato per l'esercizio in fase di creazione.

Una volta completata la traiettoria si entra nella fase finale dove, con la stessa tecnica utilizzata per il countdown si realizza un solo beep finale sempre della durata di 0.5 secondi. Per facilitare il lavoro in fase di programmazione si è scelto di affidare a questo segnale, sebbene del tutto indifferente da quelli che compongono il conto alla rovescia, un valore di flag diverso da quest'ultimi.

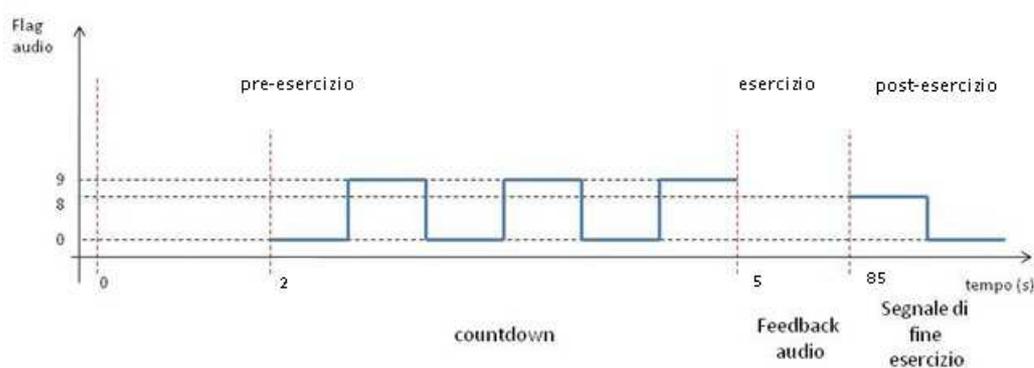


Figura 3.1: valore del flag per il tipo di audio durante le varie fasi di un esercizio

Esercizio caricato da memoria

Quando invece l'esercizio non è eseguito in real time ma viene caricato da memoria, allora non è necessaria alcuna fase pre-esercizio o post-esercizio con i relativi segnali di richiamo per il soggetto. La riproduzione interessa solamente la fase di esercizio vera e propria nella quale però verrà ristabilita la comunicazione con l'applicazione esterna per permettere di generare anche in questa modalità le stesse condizioni uditive che hanno interessato il soggetto al momento dell'esecuzione originale del test. Ciò può essere semplicemente realizzato inviando nel pacchetto di comunicazione, al campo previsto per il flag audio, il valore relativo al feedback scelto, che era stato appositamente memorizzato nella dodicesima colonna della matrice dei dati.

Un'ultimo accorgimento riguarda l'annullamento dell'audio in fase di pausa e dopo il completamento del beep di fine esercizio. Dal momento che la comunicazione con il software dei suoni rimane sempre aperta assieme all'interfaccia, si

verifica in queste situazioni la sconveniente e fastidiosa riproduzione, per un tempo indefinito, dell'ultimo segnale sonoro inviato durante l'esercizio. Per evitare ciò bisogna costruire ed inviare un pacchetto, dal contenuto informativo nullo per quanto riguarda l'esercizio ma con valore del flag pari a 0 in modo impostare il feedback nella modalità "no audio".

3.3 Comunicazione con il software dei suoni

3.3.1 Regole di creazione dei pacchetti : il protocollo OSC

L'Open Sound Control (OSC) e' un protocollo di trasmissione che permette a strumenti musicali, computer ed altri dispositivi multimedia di scambiare quello che i creatori semplificano in "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), permette di scambiare un maggior numero di dati in confronto al MIDI, dando inoltre maggiore flessibilità al tipo di dati a disposizione; tutto questo comporta la possibilità di un più alto livello di interazione tra le macchine/software. I vantaggi nei confronti del MIDI sono molteplici oltre alla differenza di velocità e di mole di dati che può essere trasmessa e 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 che è alla base delle comunicazioni è la stessa dello scambio di dati su internet e permette di inviare diversi tipi di dati. Il protocollo utilizzato per l'invio dei dati è l'UDP (user datagrams protocol).

Tipi di dati fondamentali

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-blob: un int32 che rappresenta la dimensione, seguito da un numero arbitrario di byte di dati in formato binario, seguiti a loro volta da 0-3 byte di zero per rendere il numero totale di bit multiplo di 32.

OSC Packets

I packets (più comunemente noti come pacchetti) sono l'unità fondamentale di trasmissione del protocollo OSC. L'applicazione che invia questi pacchetti prende il nome di client mentre quella che li riceve viene definita server. Tali pacchetti sono costituiti sia da dati in essi contenuti, scritti in formato binario, sia dalla dimensione di questi dati. Come già specificato precedentemente, il numero di byte che occupano deve essere multiplo di 4. Altro elemento molto importante da ricordare è che la rete, nell'atto di trasportare i pacchetti, 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 due 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 TypeTag	OSC Arguments
i	int32
f	float32
s	stringa
b	OSC-blob

Figura 3.2: OSC Type Tag

OSC Messages Un OSC Message 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 (Fig. 3.2) 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 Un “bundle” (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.

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.

Prima di tutto è necessario definire il dato da inviare, che nel caso specifico è composto da un vettore di dimensioni 1x9:

```
pd_packet = [single(xy_pat(1)) single(xy_pat(2)) single(xtar(indt))
single(ytar(indt)) single(vtar_x(indt)) single(vtar_y(indt)) single(vpat_x)
single(vpat_y) single(flag)];
```

Dove:

- *xy_pat(1)* e *xy_pat(2)*: indicano rispettivamente la posizione del soggetto rispetto all'asse x e y dello schermo
- *xtar(indt)* e *ytar(indt)*: indicano rispettivamente la posizione del target rispetto all'asse x e y dello schermo
- *vtar_x(indt)* e *vtar_y(indt)*: indicano rispettivamente la velocità del target rispetto alla direzione x e y dello schermo
- *vpat_x(indt)* e *vpat_y(indt)*: indicano rispettivamente la velocità del soggetto rispetto alla direzione x e y dello schermo
- *flag*: indica il riferimento al feedback audio percepito dal paziente (vedi Tab. 3.1)

La variabile *indt* è l'indice della posizione del vettore della traiettoria che viene graficata al momento dell'invio del pacchetto, mentre la funzione *single* 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:

```
[attr 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, mentre *attr* è una struttura che contiene i campi *size* (quanti byte sono richiesti per salvare il dato) e *align* (allineamento dei byte). I dati vengono convertiti in *single* prima di essere passati a *cstruct* in

quanto in questo modo viene rispettato l'OSC 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 ogni caso il numero totale dei byte multiplo di 4. La stringa da inviare a Pure Data, ovvero l'OSC Server, è la seguente:

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

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

3.3.2 Invio dei pacchetti: protocollo UDP

User Datagram Protocol (UDP) è un protocollo di trasporto a pacchetto, usato di solito in combinazione con il protocollo IP, e rappresenta uno dei più importanti fra la suite di protocolli Internet. E' un protocollo non orientato alla connessione, utilizzato quando l'affidabilità, il cui controllo viene richiesto ai protocolli applicativi che ne fanno uso, non è l'obiettivo primario. Le principali caratteristiche di questo protocollo possono essere riassunte nei seguenti punti:

- è un protocollo di tipo *connectionless*, ovvero suddivide il flusso di dati in entità (pacchetti) che vengono instradati in modo indipendente l'uno dall'altro, senza interazioni di ritorno fra sorgente e destinatario. L'UDP quindi non gestisce il riordinamento dei pacchetti nè la trasmissione di quelli persi, ed è perciò considerato di minore affidabilità.
- fornisce solo i servizi basilari del livello di trasporto, ovvero:
 - manipolazione delle connessioni, ottenuta attraverso il meccanismo delle porte
 - verifica degli errori mediante una checksum, inserita in un campo dell'intestazione del pacchetto

Riassumendo, il protocollo UDP è un protocollo *stateless*, ovvero non tiene conto dello stato della connessione, dunque ha meno informazioni da memorizzare

rispetto ad altri tipi di protocolli ed inoltre è molto rapido ed efficiente per le applicazioni “leggere”, come nel caso di questo studio.

Vantaggi nell'utilizzo del protocollo UDP I vantaggi nell'utilizzo di UDP sono la velocità, la minore congestione di rete rispetto a TCP (noto protocollo che svolge funzioni analoghe) in quanto non c'è scambio di pacchetti di conferma, e la possibilità di trasmettere in *multicast* (invio di un pacchetto ad un gruppo di host) e *broadcast* (invio di un pacchetto a tutti gli host di un segmento di rete).

Spedizione di pacchetti da parte dell'interfaccia

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 successivamente da inviare tramite il comando 'writepacket'.

Successivamente si crea la connessione fra la porta dell'host di destinazione e il socket UDP, per evitare di specificare di volta in volta nel comando 'writepacket' la porta di destinazione dei pacchetti:

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

In questo caso il valore 3456 identifica la porta remota che riceve il socket.

Dopo aver ultimato il dato da inviare in accordo con le regole del protocollo OSC, si procede a scrivere il dato nel buffer del socket e ad inviarlo. I comandi a cui bisogna ricorrere sono:

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

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

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

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

3.4 Software di generazione di suoni: Pure Data

Pure Data (o Pd) 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"). Un programma viene realizzato 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 altri scopi, 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.

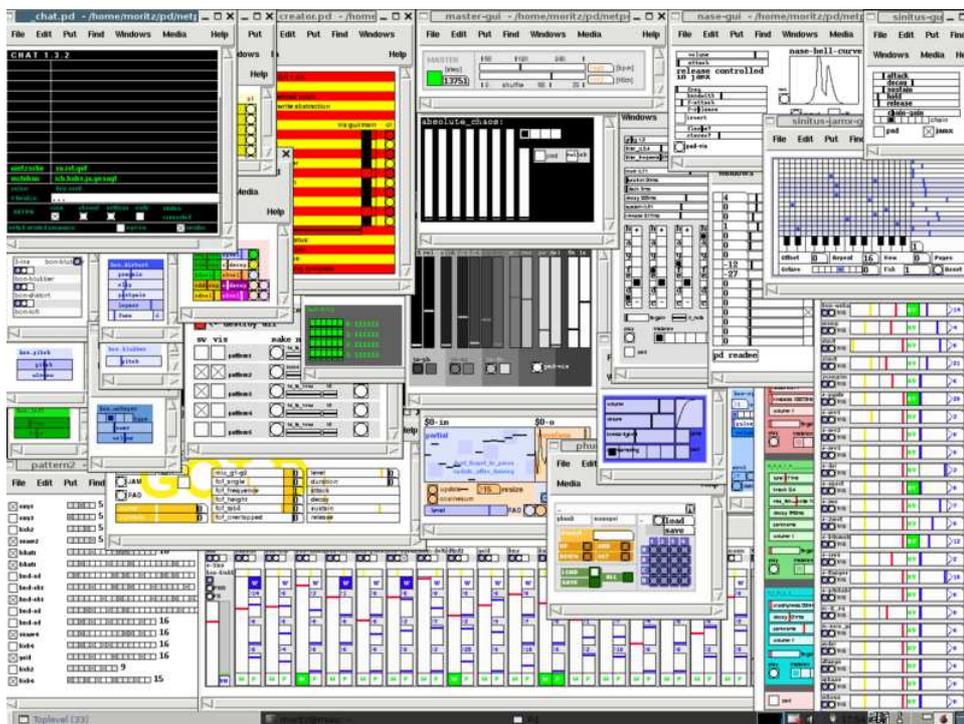


Figura 3.3: Esempio di interfaccia grafica in Pure Data

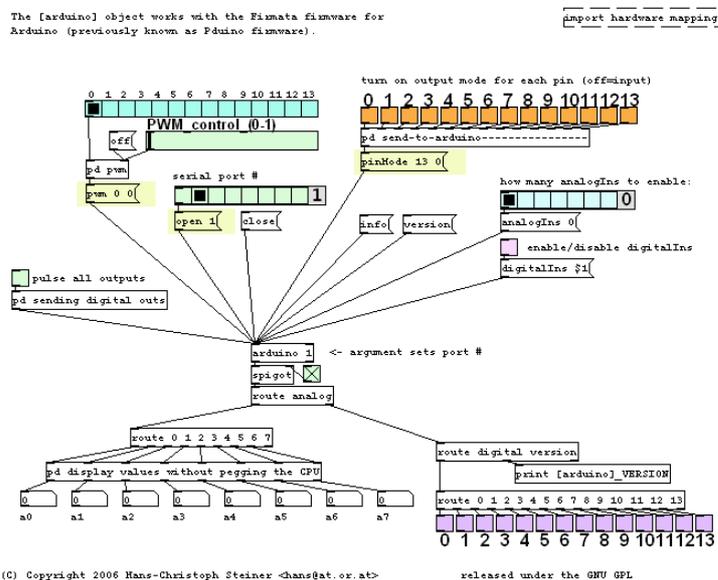


Figura 3.4: Esempio di struttura ad albero di Pure Data

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.

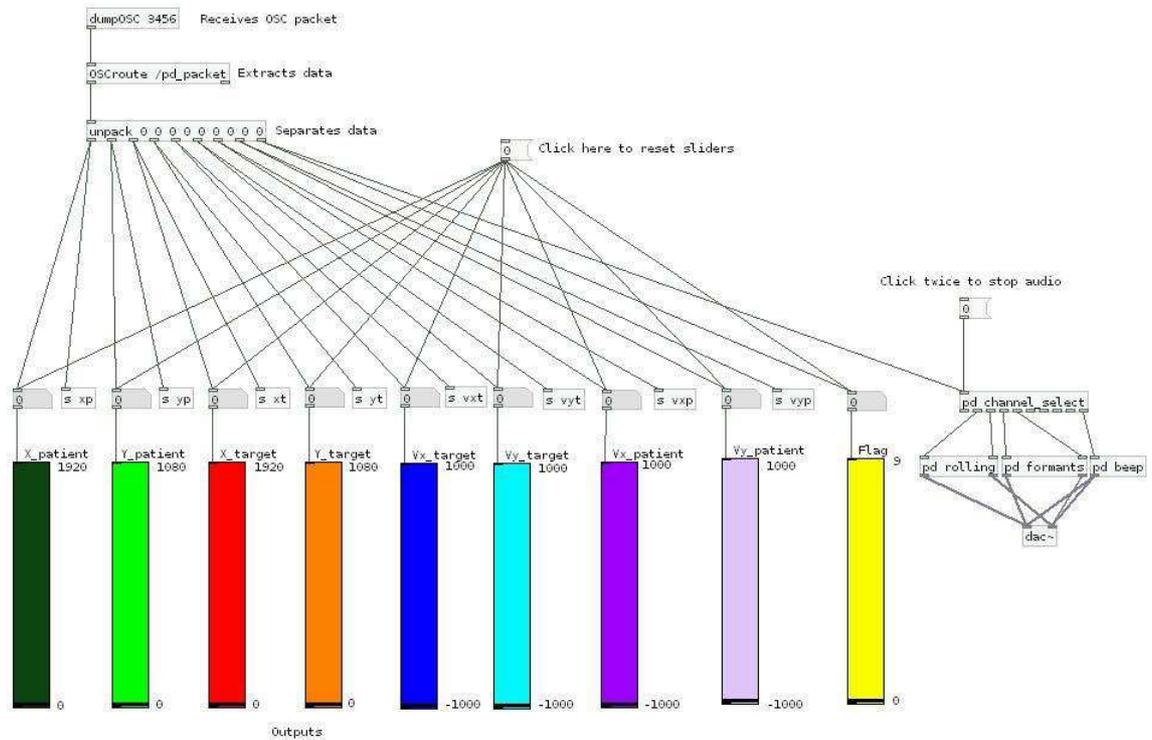


Figura 3.5: Interfaccia di Pure Data utilizzata per testare il funzionamento della mia GUI.

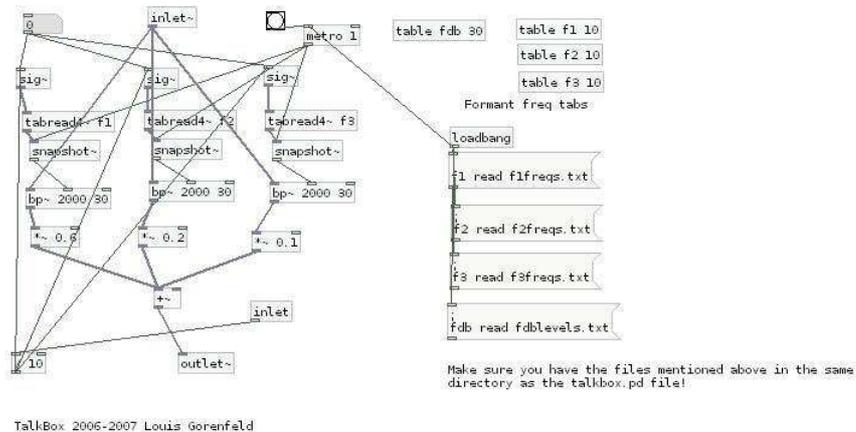


Figura 3.6: Finestra di talk box che mette in relazione tra loro i vari feedback audio

Capitolo 4

Test sperimentali

4.1 Interfaccia in modalità test

Dal momento che l'interfaccia nasce con lo scopo di essere utilizzata nella fase di test su persone, essa prevede delle soluzioni grafiche e implementative mirate a rendere più agile e semplice quelle che sono le operazioni tipiche che vengono da eseguire durante una sessione di test su un gruppo di soggetti. Inoltre la maggior parte delle scelte riguardanti le impostazioni principali sono state fatte nell'ottica di rendere ottimali le prestazioni dell'interfaccia durante un generico test.

Tutte le seguenti specifiche sono state definite in modo da rendere l'interfaccia il più possibile orientata all'esecuzione dei task motori:

- la modalità di visualizzazione di default prevede di rappresentare la finestra di esercizio su uno schermo separato, in modo da isolare anche ambientalmente il soggetto, accorgimento che si ritiene sensato in fase di test
- la frequenza alla quale lavora il timer, e quindi la frequenza con la quale si graficano gli indicatori di target e soggetto, è stata impostata a 50 Hz, valore che rappresenta il giusto compromesso fra precisione e fluidità della rappresentazione, fattore quest'ultimo fondamentale in fase di test affinché ciò che il soggetto veda sullo schermo rispecchi il più prontamente possibile l'effettivo svolgimento del task

- la durata di tutta la fase precedente l'avvio effettivo dell'esercizio e del conto alla rovescia sono state impostate rispettivamente a 5 secondi e 3 secondi, cioè valori proporzionati alla durata che si è scelta per i task (80 secondi come verrà descritto in seguito)
- la struttura dati predisposta per la memorizzazione di un esercizio prevede già tutti i campi necessari alla maggior parte delle analisi statistiche successive ai test
- il valore (fisso) per la coordinata y impostata per l'indicatore del target fa sì che esso appaia ad un'altezza alla quale la messa a fuoco dei movimenti sullo schermo da parte del paziente non risulti difficoltosa
- in un'apposita sezione dell'interfaccia viene fornito un menù dal quale si possono caricare direttamente i task pianificati per i test sui soggetti, senza dover di volta in volta andare a creare manualmente, inserendo ad uno ad uno i valori per i diversi parametri, la traiettoria di esercizio (Fig.4.1)

Impostazioni preliminari

Selezione Task prestabilito

Selezione task con feedback audio

Selezione traiettoria e rego

1. traiettoria a lunghezza fissa

Velocità: 500 [pixel / secondo]

Numero di cicli: 10

Raggio Target: 25 [pixel]

Lunghezza minima: 20 [%]

Lunghezza massima: 90 [%]

Feedback audio

Task 0: Test preliminare: prova del feedback audio
Task A: traiettoria fissa senza audio
Task B: traiettoria variabile senza audio
Task C: traiettoria fissa con audio legato alla velocità del target
Task D: traiettoria variabile con audio legato alla velocità del target
Task E: traiettoria fissa con audio legato all'errore di posizione del paziente
Task F: traiettoria variabile con audio legato all'errore di posizione del paziente

Figura 4.1: sezione dell'interfaccia dedicata alla scelta del task.

4.2 Definizione dei task

Per le sessioni di test sono stati realizzati sei tipi di task differenti combinando le tre modalità di audio scelte (no audio, audio legato alla velocità del target spazializzato, audio legato all'errore di posizione spazializzato) con i due tipi di traiettoria realizzabili (traiettoria con lunghezza dei tratti costante, traiettoria con lunghezza dei tratti variabile)

Caratteristiche comuni fra i vari task

Tutti i task, seppur diversi per tipo di traiettoria e/o tipo di audio, presentano diverse caratteristiche comuni. Ciò è particolarmente importante considerando le traiettorie con lunghezze dei tratti variabili le quali vengono generate, in maniera casuale, sempre diversa l'una dall'altra. La standardizzazione di queste, attraverso la scelta di alcuni parametri comuni, permette di assumere queste traiettorie comparabili in termini di livello di difficoltà, evitando di influenzare in maniera rilevante le successive analisi sui dati.

Tutti i task sono costituiti da una serie di movimenti orizzontali in direzione alternata che inizia e termina a nella stessa posizione di centro schermo.

Ogni task presenta una durata di 80 secondi, ed è preceduto da una fase iniziale, durante la quale il target rimane fermo a centro schermo, della durata di 5 secondi, di cui gli ultimi 3 sono interessati dall'esecuzione del conto alla rovescia.

Ogni task si compone di 28 tratti di movimento orizzontale, 26 di questi possono essere definiti completi poichè il movimento interessa tutta la percentuale della larghezza dello schermo che si rende disponibile alla traiettoria, gli altri 2, cioè il primo e l'ultimo, possono essere definiti dei semi tratti in quanto portano l'indicatore del target dalla posizione iniziale al primo estremo (tratto 1) e dall'ultimo estremo della traiettoria nuovamente nella posizione iniziale (tratto 28). I tratti 1 e 28 vengono percorsi con velocità media pari alla metà di quella stabilita per i tratti ordinari (dal 2 al 26).

La velocità media dei tratti 2-26 è stata impostata a 400 pixel/secondo per ogni task.

La lunghezza dei tratti nei task con traiettoria a lunghezza costante è pari al 60% della larghezza massima. La lunghezza dei tratti per i task con traiettoria a lunghezza variabile oscilla tra il 20% e il 90% della larghezza massima.

La dimensione del raggio degli indicatori è stata fissata a 25 pixel.

Di seguito sono riportate le descrizioni dei sei tipi di task predisposti per i test:

1. Task A : traiettoria con tratti a lunghezza fissa privo di audio
2. Task B : traiettoria con tratti a lunghezza variabile privo di audio
3. Task C : traiettoria con tratti a lunghezza fissa con feedback uditivo spazializzato sulla velocità del target
4. Task D : traiettoria con tratti a lunghezza variabile con feedback uditivo spazializzato sulla velocità del target
5. Task E : traiettoria con tratti a lunghezza fissa con feedback uditivo spazializzato sull'errore di posizione
6. Task F : traiettoria con tratti a lunghezza variabile con feedback uditivo spazializzato sull'errore di posizione

	Lunghezza costante	Lunghezza variabile	Feedback audio
A	✓		0
B		✓	0
C	✓		2
D		✓	2
E	✓		4
F		✓	4

Tabella 4.1: Riassunto dei diversi tipi di task creati. Si noti che nella colonna del feedback audio è stato specificato direttamente il valore per il flag secondo la convenzione adottata (vedi 3.1).

4.3 Popolazione

I test sono stati effettuati su una popolazione di 20 soggetti di età media 24.75 anni e range di variabilità 21-29 anni. La popolazione considerata era composta da 11 uomini e 9 donne tutti destrosi.

Nessuno dei soggetti sottoposti al test presentava disabilità di alcun genere ed entità, o tali da compromettere la qualità dell'esecuzione del test. Lo studio su soggetti non disabili è stata obbligata da ragioni di comodità e da problemi di tempistica, tuttavia i risultati tratti da questa popolazione possono essere utilizzati per caratterizzare la normale risposta di un soggetto sano ad una determinata tipologia di feedback audio e possono rappresentare la base per confronti futuri con altri tipi di soggetti.

4.4 Protocollo di 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 veniva fatto accomodare dietro ad una scrivania provvista di schermo, tavoletta grafica e cuffie.

Schermo Il soggetto fissava l'andamento in tempo reale dell'esercizio su uno schermo Full HD di risoluzione 1080x1920 posizionato al centro della scrivania, con alle spalle una parete bianca, completamente libera.

Tavoletta grafica In sostituzione ad un vero e proprio robot riabilitativo per l'acquisizione dei movimenti dei soggetti si è ricorso ad una tavoletta grafica appoggiata sul piano orizzontale della scrivania. I soggetti impugnavano un puntatore (a forma di stilo) con il quale tracciavano i loro inseguimenti sulla tavoletta grafica.

Cuffie La fruizione del feedback uditivo avveniva per mezzo di un sistema a cuffie. L'utilizzo delle cuffie presenta alcuni svantaggi se comparato ad altri sistemi audio (ad esempio quelli stereo): esse sono invasive e possono essere scomode da

indossare per lunghi periodo di tempo, inoltre non presentano una risposta in frequenza equalizzata (piatta nella banda passante). Tuttavia presentano due grandi vantaggi che sono quelli di eliminare il riverbero e il rumore di sottofondo proprio dello spazio sonoro circostante, e di rendere possibile l'invio di segnali distinti ad ogni orecchio. Inoltre, banale ma da non trascurare, è il vantaggio in termini di costi e flessibilità granatito dall'utilizzo di un semplice sistema a cuffie per la resa dell'audio.



Figura 4.2: Esempio di test

La sessione di test consisteva nella successione dei sei task riabilitativi, intervallati l'uno dall'altro da un periodo di pausa di rilassamento, proposti ad ogni soggetto in maniera randomizzata al fine di evitare che i dati da analizzare fossero influenzati dall'ordine con cui sono stati eseguiti gli esercizi o da normali effetti di learning che si possono verificare nell'esecuzione di semplici movimenti ripetitivi. Infatti appare lampante che, in proporzione, l'inseguimento nel primo task del test sarà sempre qualitativamente peggiore dell'inseguimento relativo all'ultimo task del test.

Il soggetto veniva istruito su postura e comportamenti corretti da tenere durante la fase di test.

L'obbiettivo che veniva richiesto al soggetto per l'esecuzione del test era semplicemente quello di seguire, al meglio delle sue possibilità, il movimento dell'indicatore circolare di colore rosso rappresentato a video.

Prima dell'inizio della successione di task, al soggetto veniva presentato un task di prova, senza alcuna traiettoria per il target, nel quale però il soggetto

poteva prendere confidenza con la tavola grafica, valutandone velocità di risposta, fluidità nella rappresentazione, sensibilità del tocco ed altre proprietà simili, al fine di intraprendere la fase di test con almeno un bagaglio di minima sintonia con lo strumento di lavoro.

Progressivamente all'esecuzione dei test è stata riempita una tabella riassuntiva (Fig. 4.3) contenente tutte le informazioni riguardo al soggetto, alla cronologia dei task, e ad eventuali situazioni singolari verificatesi nell'esecuzione del test, specificando il task e, se possibile, l'istante di tempo nel quale tale irregolarità ha avuto luogo.

PIANIFICAZIONE TASKS - Generazione di feedback sonoro nell'esecuzione di task motori

	SOGGETTO	TASK	ETA'	SESSO	BRACCIO	IRREGOLARITA' NEI TEST					
						Distrazione sogg.	Mano alzata	Scatti pc		Interruzione task	
								SI	Istante scatto	SI	Istante interruzione
1	Romani Daniele	C E B F A D	24	M	dx	Ingresso persona nella stanza		B A D	68s 70s 37s		
2	Benedetti Giulio	E C B F A D	24	M	dx	Mano d'appoggio staccata (C-55s)		C	70s		
3	Pasqualin Alice	D B C F A E	22	F	dx	Perdita di controllo sul puntatore (A-115s)		C A	48s 37s	X	cavo di alimentazione dello schermo staccato
4	Costanzi Enrico	E C F B A D	22	M	dx	Squillo del telefono (F-40s)		A D	105s 3s		
5	Carli Andrea	B C A F E D	26	M	dx	Ingresso persona nella stanza (B-15s)		C	2s		
6	Frison Marco	C A E D B F	22	M	dx			C D F	30s 47s 15s		
7	Caramma Lea	F E D B C A	21	F	dx	Cambio posizione mano d'appoggio (C-20s)		E D A	40s 52s 40s		
8	De Faveri Valentino	D C E F B A	23	M	dx			D F F	18s 2s 57s		
9	Sartorato Paolo	A E B F D C	22	M	dx			A E	4s 32s		
10	Nardo Alice	F D B E A C	22	F	dx	Il soggetto ha stamutato (D-15s)		D	40s		
08-lug											
11	Milani Matteo	B D F A C E	23	M	dx	Ingresso persona nella stanza (A-30s)		D E	40s 37s		
12	Maroni Ida	C F E B D A	22	F	dx						
13	Zanata Chiara	F E D A C B	22	F	dx		X (F)	C	70-75s		
14	Pavan Beatrice	D F B A E C	21	F	dx	Squillo del telefono (A-40s)		B	52s		
15	Zampicini Andrea	F A B D C E	29	M	dx	Ingresso persona nella stanza (B-75s)		C	50s		
16	Carraro Martina	C A F D E B	22	F	dx	Molto vicino al bordo sx della tavoletta (F)		D	43s		
17	Cardin Silvia	B D C F A E	22	F	dx			E	50s		
18	Garbujo Alessio	E F C A D B	22	M	dx			F	54		
19	Nicoli Paola	E B C A F D	26	F	dx					X	Il task E è stato ripetuto perché pc bloccato
20	Smania Giovanni	D F A C E B	22	M	dx						

Figura 4.3: Tabella finale con tutti i dati riguardo i soggetti e le anomalie durante i task.

4.5 Memorizzazione dei task

Per semplificare le operazioni di elaborazione e analisi sui dati memorizzati per ogni task, si è cercato di scegliere una denominazione per i file che fosse concisa e soprattutto uniforme in tutta la durata dei test. I dati di ciascun soggetto sono stati infatti salvati all'interno della stessa cartella con la stringa *TaskLetter-aTask-CognomeSoggetto*.

Ogni file contiene la struttura denominata *exercise_parameters* contenente tutte le informazioni e i dati utili in fase di analisi per quello specifico task di quel particolare soggetto. In particolare la struttura *exercise_parameters* prevede i seguenti campi:

- *data* : matrice contenente i dati relativi all'esecuzione dell'esercizio (vedi 2.4).
- *velocità_target* : velocità media del target nei tratti del task in pixel al secondo
- *numero_iterazioni* : numero di tratti di cui si compone ogni esercizio
- *lunghezza_minima* : percentuale sulla massima dimensione dello schermo per calcolare la minima lunghezza consentita ad un tratto del task
- *lunghezza_massima* : percentuale sulla massima dimensione dello schermo per calcolare la massima lunghezza consentita ad un tratto del task
- *raggio_target* : dimensioni in pixel del raggio del target
- *dimensioni_schermo* : vettore [*larghezza altezza*] delle dimensioni in pixel dello schermo sul quale è avvenuta l'esecuzione del task
- *traiettoria_selezionata* : intero indicante il tipo di traiettoria scelta (1 per lunghezza fissa, 2 per lunghezza variabile)
- *descrizione_esercizio* : stringa con la descrizione del tipo di traiettoria scelta
- *pianificazione* : intero indicante il tipo di funzione scelta per la pianificazione della traiettoria

- n : intero indicante la posizione ricoperta dal task nella lista randomizzata per l'ordine di esecuzione del relativo soggetto

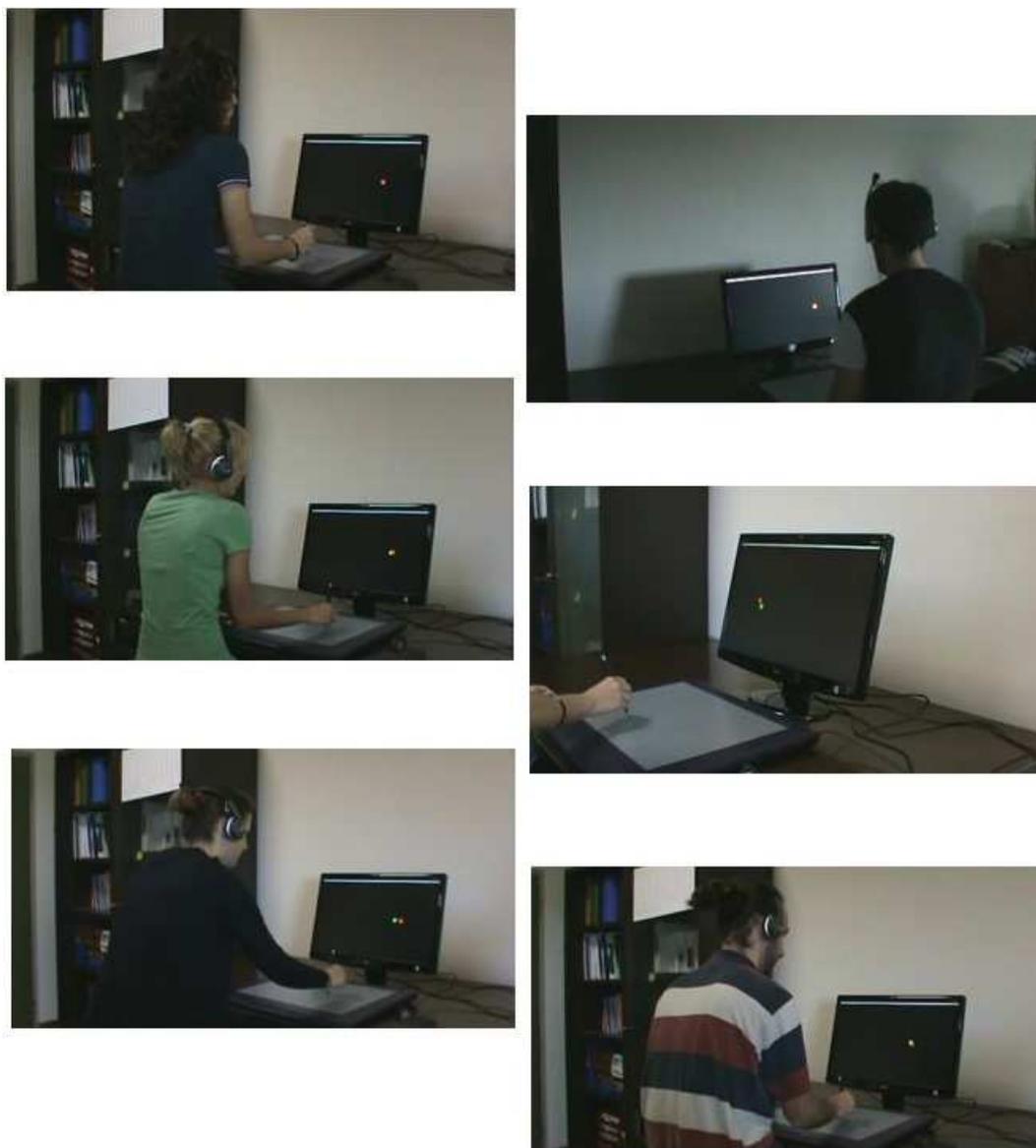


Figura 4.4: Alcune fotografie dei test effettuati in data 08/07/2010.

Capitolo 5

Elaborazione dei dati e analisi statistica

Come più volte sottolineato, lo scopo di questo studio è quello di valutare se, in che modo e in che entità l'aggiunta di un feedback audio vada ad incidere sull'esecuzione di semplici task motori volti al recupero delle funzionalità dell'arto superiore di un soggetto che ha subito un ictus.

Dal momento che il risultato che si vuole evidenziare sono le modifiche apportate dalla presenza di un appropriato feedback audio 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.

5.1 Elaborazioni preliminari sui dati

Come primo passo di elaborazione vengono eseguite una serie di operazioni preliminari per predisporre i dati relativi ai task in una forma che sia adatta al tipo di analisi che si andrà ad effettuare.

5.1.1 Rimozione delle righe nulle

La matrice dei dati contenuta nella struttura *exercise_parameters* generalmente presenta, dopo l'ultimo campione della traiettoria un insieme più o meno lungo di righe in cui tutte le celle sono impostata al valore nullo. Ciò è conseguenza del fatto che la matrice dei dati viene creata specificandone anche la dimensione, prima dell'avvio del test quando non si sa ancora di quanti campioni si comporrà la traiettoria rappresentata. Per rimuovere tali righe prive di significato basta imporre che la componente della dimensione della matrice riferita al numero di righe sia pari all'indice dell'ultima riga non vuota, come viene fatto di seguito:

```
last = find(exercise_parameters.data(2:end, 1) == 0, 1, 'first');  
if ~(isempty(last))  
    exercise_parameters.data = exercise_parameters.data(1 : last, :);  
end
```

dove si cerca l'indice della prima riga (escludendo la prima) che presenta un valore nullo nella prima colonna, quella dei tempi, condizione rivelante l'assenza di dati, e si utilizza tale indice per limitare la dimensione della matrice dei dati.

5.1.2 Calcolo della velocità della velocità del soggetto

A differenza della velocità del target la quale è nota con esattezza già in sede di pianificazione, poichè ottenuta come derivata parziale secondo la direzione x e y della funzione polinomiale di quinto grado utilizzata per la traiettoria del target, la velocità puntuale del soggetto nelle componenti x e y deve essere valutata in un secondo momento ricorrendo a metodi di approssimazione numerica. L'inseguimento del soggetto infatti non può essere ricondotto ad una funzione matematica esatta da derivare.

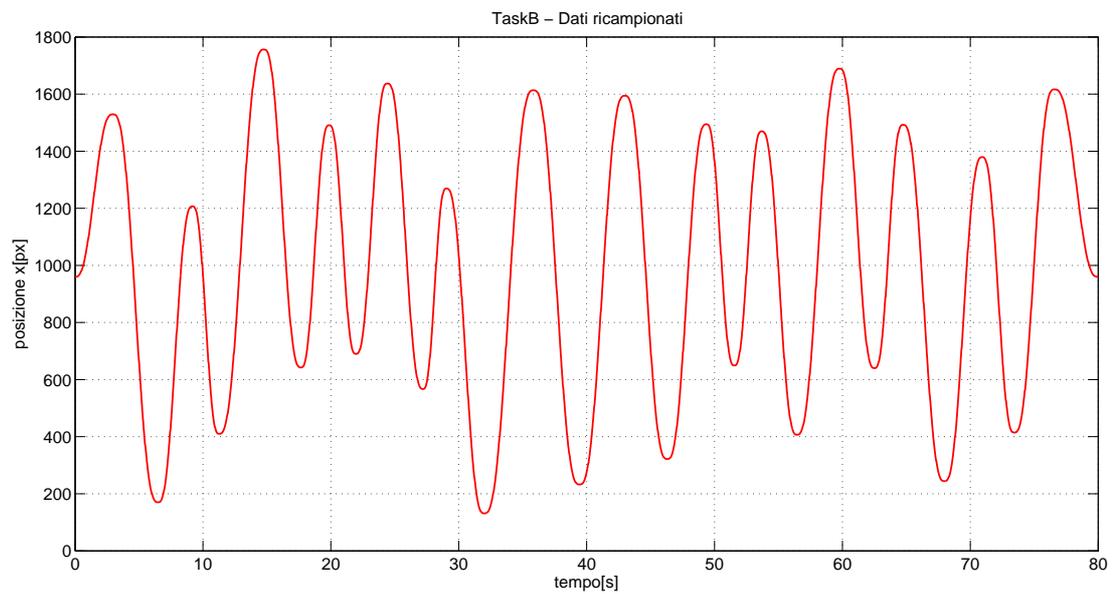


Figura 5.1: Tracciato della traiettoria del target in un esercizio a lunghezza variabile: ogni tratto è riconducibile ad un polinomio di quinto grado.

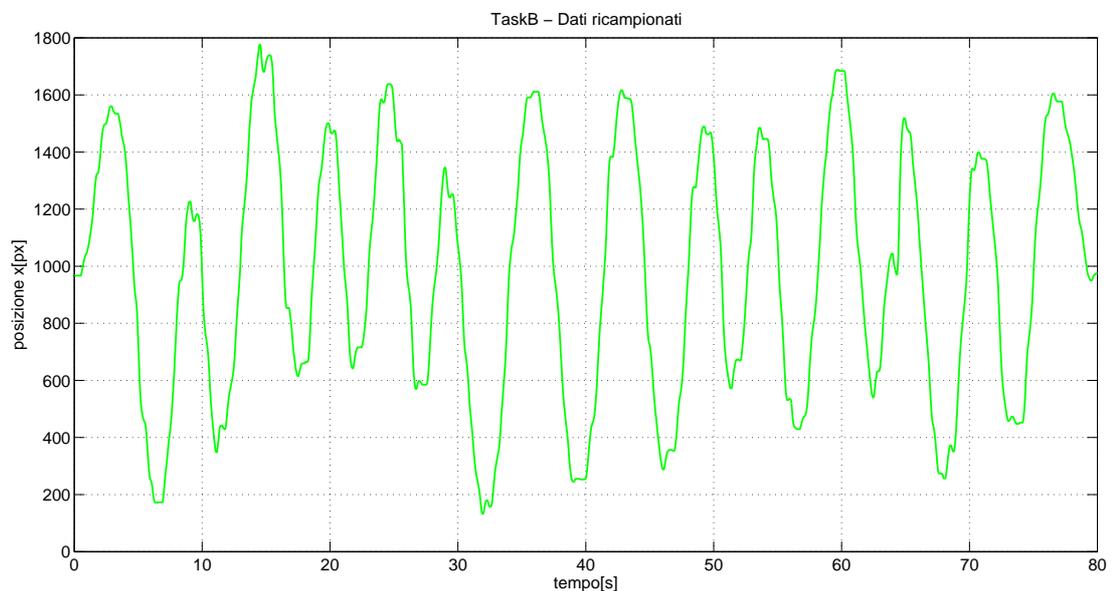


Figura 5.2: Tracciato della traiettoria del soggetto per lo stesso esercizio: non è possibile dare un'espressione matematica ad ogni tratto.

Il metodo più semplice ed immediato per calcolare la velocità del soggetto

secondo la direzione x e y del movimento è quello di derivare numericamente il vettore delle posizioni rispetto al vettore dei tempi. La derivata numerica in ogni punto del vettore viene calcolata come rapporto differenziale fra due campioni adiacenti della traiettoria. Siano:

- x_s vettore delle posizioni in x del soggetto
- t vettore dei tempi (istanti di campionamento)

Se si suppone che il tempo t_1 corrisponda nel vettore t al tempo relativo alla posizione k , allora si può definire derivata approssimata della posizione k del vettore x_s rispetto al vettore t , all'istante di tempo t_1 , la relazione seguente:

$$\dot{x}_s(k) = \frac{x_s(k) - x_s(k-1)}{t(k) - t(k-1)}$$

Per rendere applicabile tale formula a tutte le posizioni del vettore (anche a quelle iniziale e finale), si è fatto ricorso ad una elaborazione di questa idea, sfruttando il metodo di interpolazione.

Interpolazione lineare di una serie di dati

In analisi numerica, per interpolazione si intende un metodo per individuare nuovi punti del piano cartesiano a partire da un insieme finito di punti dati, nell'ipotesi che tutti i punti si possano riferire ad una funzione $f(x)$ di una data famiglia.

Sia data una sequenza di n numeri reali distinti x_k chiamati nodi e per ciascuno di questi x_k sia dati un secondo numero y_k . Il metodo di interpolazione si propone di individuare una funzione f di una certa famiglia tale che sia:

$$f(x_k) = y_k \text{ per } 1, \dots, n$$

In particolare, l'interpolazione lineare per ogni coppia di punti consecutivi dati, denotati (x_a, y_a) e (x_b, y_b) , definisce come funzione interpolante nell'intervallo $[x_a, x_b]$ la funzione seguente:

$$f(x) = \frac{x - x_b}{x_a - x_b} y_a - \frac{x - x_a}{x_a - x_b} y_b$$

con un errore di interpolazione proporzionale al quadrato della distanza fra i punti dati.

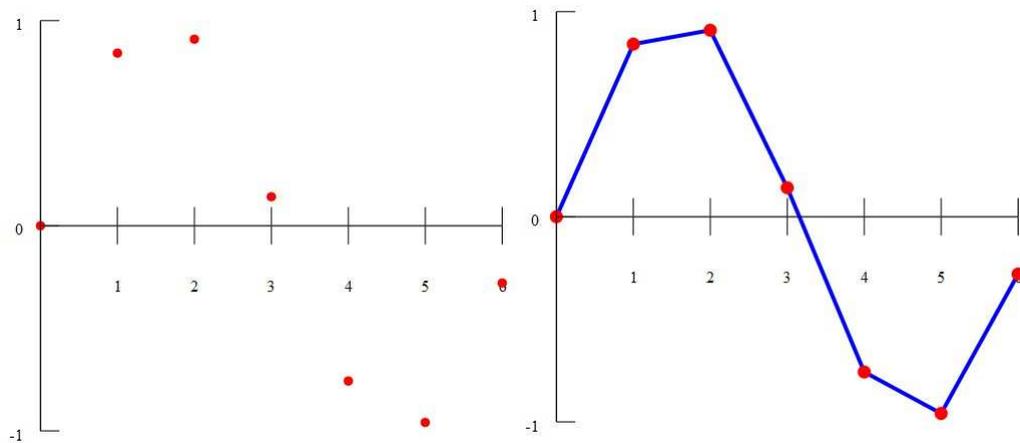


Figura 5.3: Esempio di interpolazione lineare.

Per la realizzazione dell'interpolazione lineare MatLab fornisce un'apposita funzione:

$$\text{interp1}(x, y, xi, 'linear', 'extrap')$$

dove x e y sono i vettori che compongono la coppia di dati, e restituisce i valori della funzione interpolata calcolati nei punti del vettore xi , utilizzando la funzione lineare per l'interpolazione, la quale viene eseguita anche su valori esterni all'intervallo dei dati (Fig.5.4).

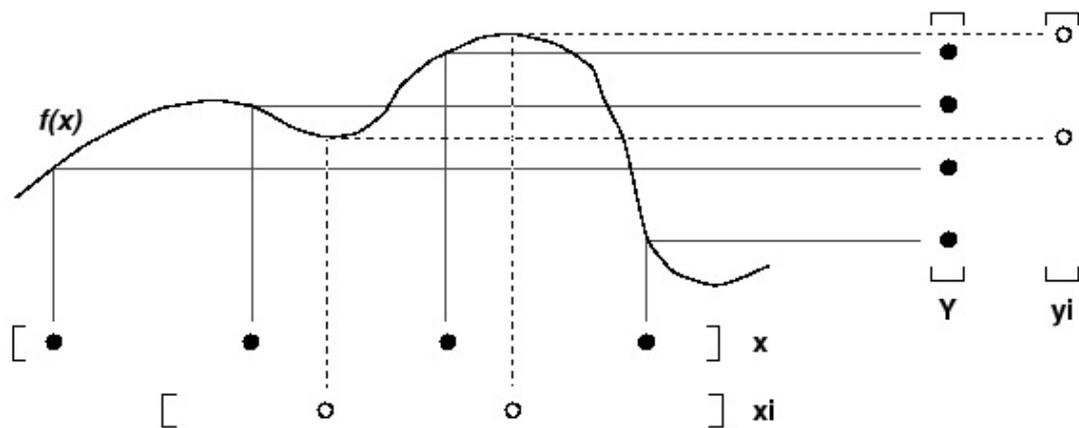


Figura 5.4: Rappresentazione del modo di operare della funzione interp.

Per il calcolo della velocità del paziente in x e in y ci serviamo delle funzione `interp1` di MatLab. Con tale funzione si costruiscono due vettori contenenti i valori interpolati per la posizione del paziente (in x o in y a seconda della componente che si sta considerando) sul vettore dei tempi traslato in ritardo ed in anticipo di una quantità piccola dx . Il rapporto incrementale viene computato eseguendo le differenze fra posizioni corrispondenti dei due vettori e dividendo per una quantità doppia dell'incremento dx , pari alla distanza fra i due punti su cui si calcola la differenza.

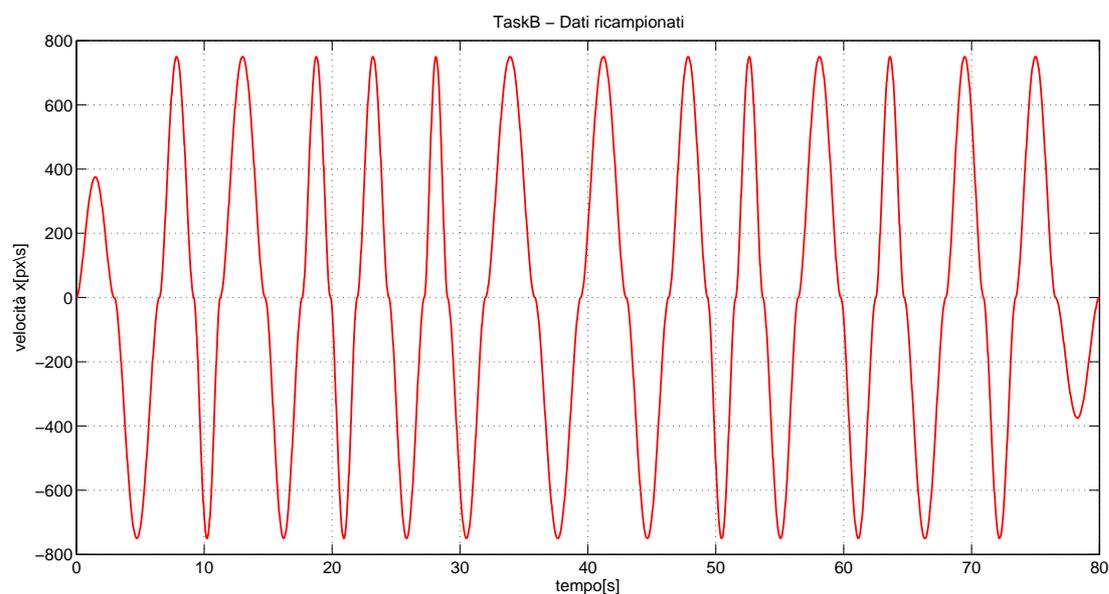


Figura 5.5: Tracciato della velocità in x del target ottenuta come derivata della funzione di pianificazione.

La velocità del soggetto risente ovviamente delle approssimazioni insite nel metodo utilizzato per il suo calcolo, in particolare si verifica lo scomodo inconveniente di brevi picchi piuttosto marcati di velocità. Tali imprecisioni, anche se lievi dal punto di vista del computo generale degli errori, possono essere attenuate attraverso un'opportuna successione di interpolazioni più "lasche".

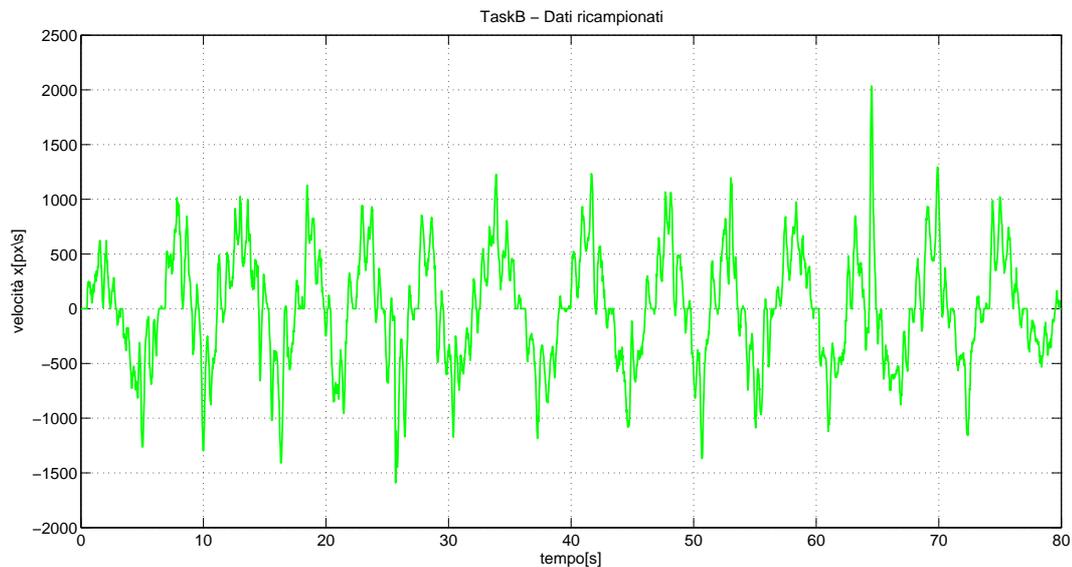


Figura 5.6: Tracciato della velocità in x del soggetto per lo stesso task riportato di sopra, ottenuta attraverso l'approssimazione numerica descritta.

5.1.3 Irregolarità del timer

Ricampionamento dei dati

Come già descritto, per garantire la corrispondenza fra tempi di pianificazione e tempi di esecuzione durante la visualizzazione dell'esercizio, si fa ricorso al calcolo degli istanti di tempo reali, attraverso l'uso del cronometro, dal momento che il timer di MatLab non è in grado di rispettare la temporizzazione impostata attraverso la sua proprietà "period" (vedi 2.3.1). Questa fatto comporta però che i campioni relativi all'esercizio non siano equidistanziati nel tempo, condizione che può essere causa di errori nel calcolo di alcuni parametri di errore. Per ovviare a questo problema si predispone nella struttura dati di ogni task memorizzato un nuovo campo, denominato *data_resamp*, nel quale inserire una copia della matrice dei dati dell'esercizio dove però le colonne sono state ricampionate su un nuovo vettore dei tempi in cui l'intervallo temporale fra un campione e l'altro è costante e pari al valore teorico di 0.02 secondi impostato nella relativa proprietà del timer. Il ricampionamento delle colonne sul vettore dei tempi "ideale" fa uso sempre della funzione *interp1*.

Scatti del timer e intervalli di riassetamento

Durante l'esecuzione di test sui soggetti si è notato che con una certa frequenza, circa un paio di volte per ogni task, la riproduzione dell'esercizio "scattava", ovvero il target sembrava bloccarsi per un intervallo di tempo nell'ordine del decimo di secondo, ma sufficiente a creare una discontinuità piuttosto percepibile nella visualizzazione al ripristino del normale funzionamento, come una sorta di blackout in cui il processo rimane congelato. Anche questo problema può probabilmente essere ricondotto al modo di operare del timer, non del tutto ottimale.

Dal momento che il verificarsi di queste situazioni, seppure piuttosto saltuarie, risultava essere una fonte di errore non desiderata e non "fisiologica" per un normale inseguimento da parte del paziente il quale, per brevi istanti di tempo, si trovava privo dell'esatto feedback visivo, è stata realizzata una funzione che durante il calcolo dei vari parametri di errore è in grado di individuare questi momenti critici e di non considerare i dati relativi all'inseguimento in un intervallo di tempo, di durata impostabile, successivo a tali eventi anomali.

L'individuazione degli "scatti" avviene per ispezione della colonna dei tempi. Ogni tanto infatti si incontrano campioni distanziati nel tempo da un intervallo troppo ampio se considerato in relazione al periodo di campionamento medio che è di poco superiore ai 0.02 secondi (Fig. 5.7 e).

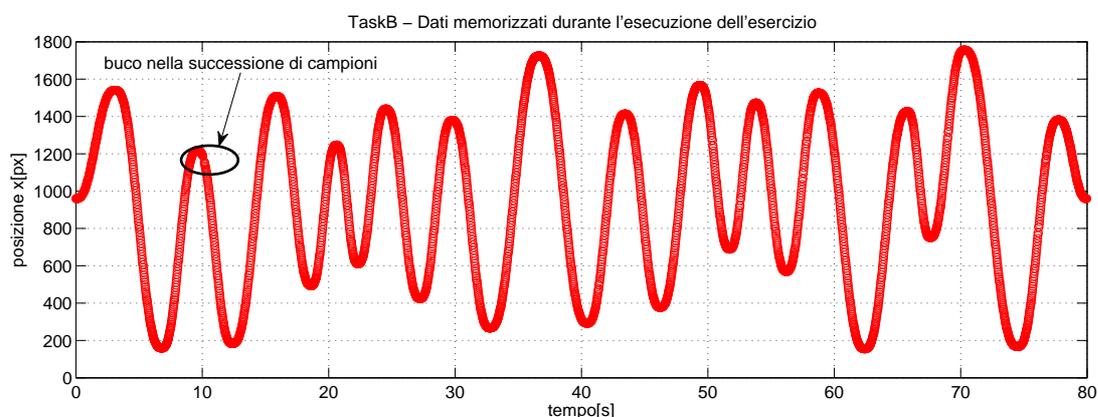


Figura 5.7: La rappresentazione dei campioni permette di individuare già dal grafico del tracciato la presenza degli scatti.

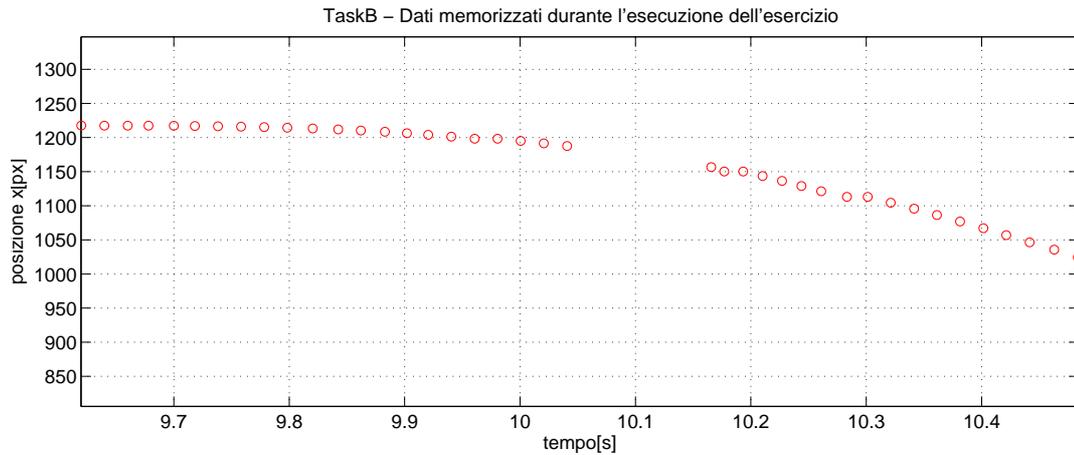


Figura 5.8: Dettaglio del profilo. Si nota la distanza temporale fra i campioni interessati dallo “scatto”.

5.2 Tipi di errore calcolati

La valutazione sulla qualità dell’inseguimento di ogni task viene fatta calcolando che valore assume un certo parametro d’errore.

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.

5.2.1 Distanza media

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.

$$\varepsilon = \frac{1}{N} \sum_{k=1}^N \frac{\sqrt{(x_s(k) - x_t(k))^2 + (y_s(k) - y_t(k))^2}}{r}$$

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 rappresentando il modulo di un vettore (vettore differenza di posizione) il risultato che fornisce questo parametro non potrà mai essere negativo e sarà indipendente dal segno delle differenze di posizione lungo gli assi x e y , cioè tratterà allo stesso modo equivalenti condizioni di ritardo o anticipo del soggetto rispetto al target.

A titolo d'esempio si riporta il grafico della distanza media calcolata in alcuni cicli di un inseguimento, costruito a posteriori, in cui il soggetto durante il movimento compie continue rotazioni di raggio costante attorno al target (Fig. 5.9).

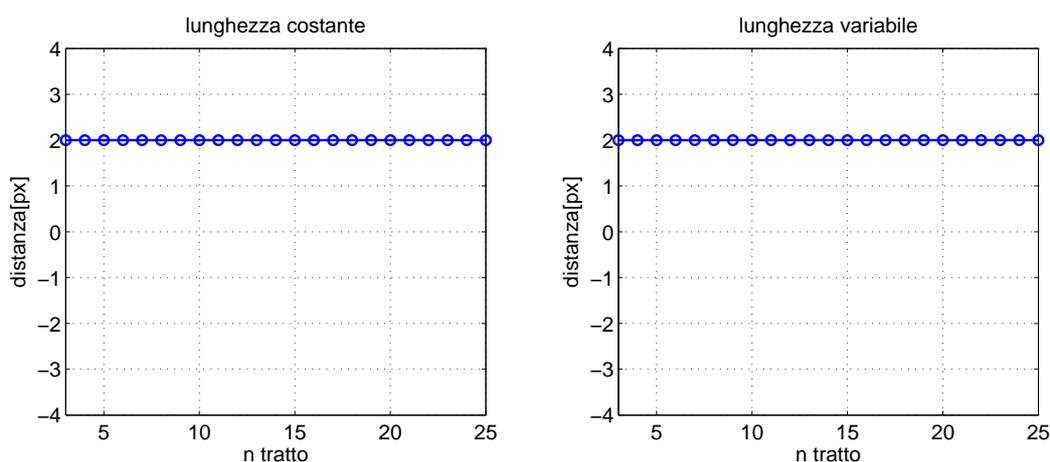


Figura 5.9: Distanza media per i cicli 3-25 dei task a lunghezza fissa e variabile con soggetto che insegue il target ruotandogli attorno con raggio costante.

La distanza rimane costante in ogni ciclo ed è di valore pari a quello scelto per il raggio che definisce l'ampiezza della rotazione attorno al target.

5.2.2 Media del modulo della velocità relativa

Per velocità relativa si intende il vettore risultante dalla differenza vettoriale fra velocità del soggetto e velocità del paziente, valutate per ogni istante di campionamento:

$$\vec{v}_{rel} = \vec{v}_s - \vec{v}_t$$

di questo vettore se ne valuta il modulo, e si media il valore risultante sul numero di campioni considerato, secondo la formula:

$$\varepsilon = \frac{1}{N} \sum_{k=1}^N \sqrt{(v_{x,s}(k) - v_{x,t}(k))^2 + (v_{y,s}(k) - v_{y,t}(k))^2}$$

dove N rappresenta il numero di campioni considerati.

Per questo tipo di errore non viene eseguita nessuna normalizzazione poichè non esiste una grandezza tipica dell'esercizio che renda significativa l'operazione di normalizzazione.

L'interpretazione che si può dare per questo tipo di errore è piuttosto immediata. Esso dà una misura di quanto differenti sono, in valore assoluto, le velocità di soggetto e target istante per istante. Ad esempio se consideriamo un inseguimento del target costruito con ritardo (o anticipo) temporale costante otteniamo un errore medio sicuramente non nullo (Fig. 5.10), se viceversa l'inseguimento è costruito con un ritardo (o anticipo) costante non nel tempo ma nello spazio questo tipo di parametro dà un valore nullo di errore, come se l'inseguimento fosse stato perfetto, ed in effetti così è se si considera solo l'aspetto "velocità" (Fig. 5.11). Le leggere fluttuazioni di questo tipo di errore nei vari tratti del task a lunghezza costante, nei quali dovrebbe rimanere costante nel primo caso e nullo nel secondo, sono dovute alle approssimazioni intrinseche presenti nel computazione della velocità del soggetto.

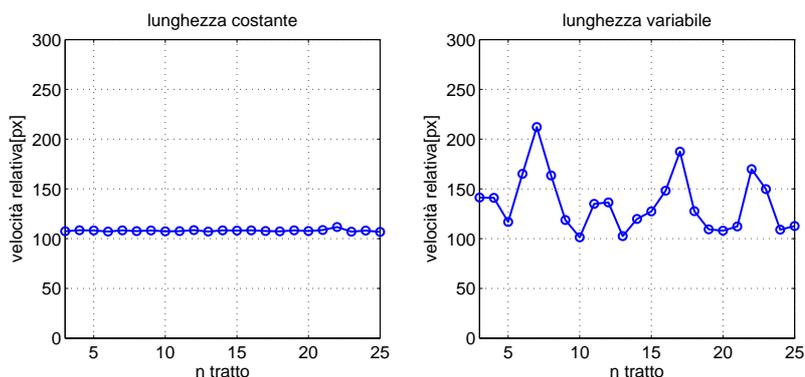


Figura 5.10: Media del modulo della velocità relativa nei tratti 3-25 di un task con un inseguimento a ritardo temporale costante.

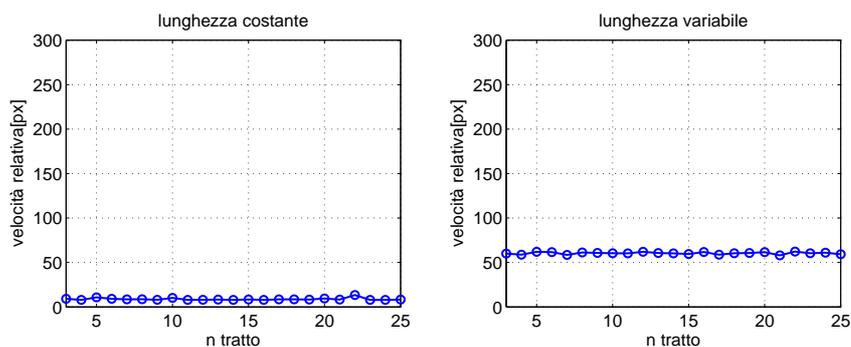


Figura 5.11: Media del modulo della velocità relativa nei tratti 3-25 di un task con un inseguimento a ritardo nella direzione x costante.

5.2.3 Integrale del modulo della velocità relativa

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

$$\int_{t_1}^{t_2} |\vec{v}_r| dt$$

con $\vec{v}_r = \vec{v}_s - \vec{v}_t$ vettore della velocità relativa.

La valutazione dell'integrale avviene in maniera approssimata sommando i contributi relativi ad ogni campione pesato con l'intervallo di campionamento, che nel caso della matrice di dati ricampionata è costante e pari a 0.02 s. Il calcolo

di questo parametro di errore consiste quindi nell'implementazione del metodo dei rettangoli per il calcolo numerico dell'integrale.

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

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

Questo parametro di errore misura lo spazio percorso, lungo un percorso rettilineo, da un punto il cui vettore velocità abbia modulo pari a quello del vettore velocità relativa (vettore differenza della velocità soggetto con la velocità target), e pertanto da un' indicazione sullo spazio percorso inutilmente in ogni tratto dal soggetto, indipendentemente dall'accuratezza con cui l'esercizio è stato eseguito, poichè la valutazione dello spazio è ottenuta per via integrativa dalla velocità. Il risultato fornito dall'integrale viene scalato sulla lunghezza del tratto percorso dal target.

Di seguito sono rappresentati gli errori per questo tipo di parametro per un task con inseguimento con rotazione del soggetto attorno al target (Fig. 5.12), e per un task con inseguimento con oscillazione del soggetto attorno al target solo nella direzione x di movimento (Fig. 5.13). Come mostrato dai grafici si vede che l'errore nel primo caso risulta superiore rispetto al secondo, in quanto il movimento di rotazione sul piano porta il soggetto a ricoprire una traiettoria di lunghezza decisamente superiore a quella relativo ad un inseguimento con sola oscillazione rispetto al moto in x. In particolare se i due tipi di rotazione avvengono con la stessa velocità angolare si può verificare che per ogni periodo di rotazione nel primo caso si percorre una traiettoria in cui la lunghezza della strada aggiuntiva percorsa è pari alla lunghezza della circonferenza delle rotazioni, mentre nel secondo la lunghezza della strada supplementare è pari al diametro della circonferenza che verrebbe descritta nel caso di rotazione completa.

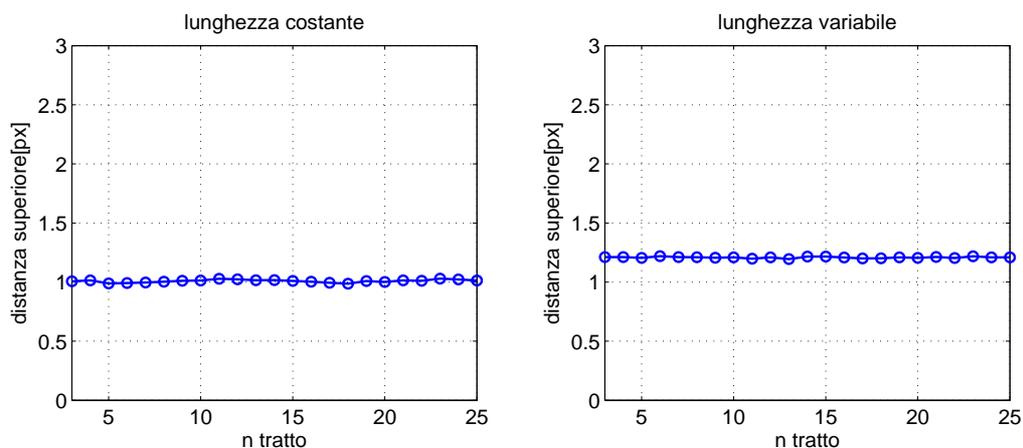


Figura 5.12: Integrale del modulo della velocità relativa nei tratti 3-25 di un task con inseguimento a rotazione del soggetto attorno al target.

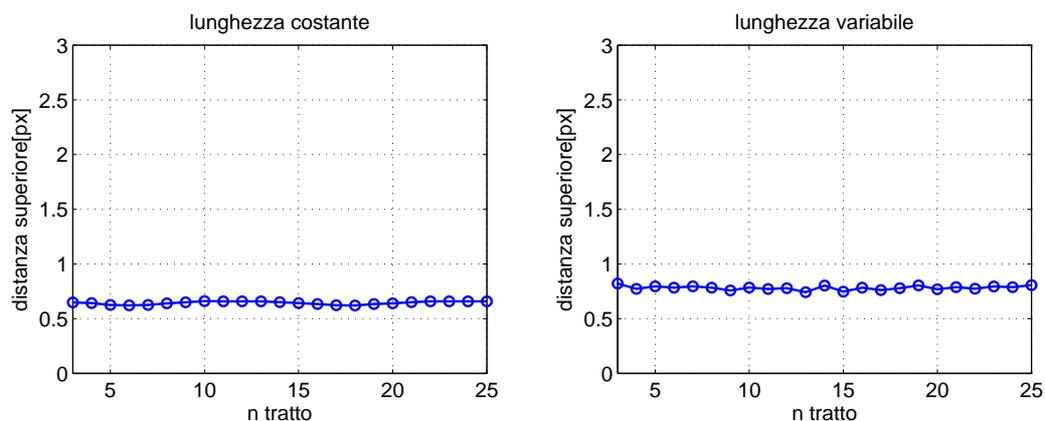


Figura 5.13: Integrale del modulo della velocità relativa nei tratti 3-25 di un task con inseguimento ad oscillazione del soggetto attorno al target solo nella direzione x di movimento.

Per meglio comprendere il significato di questo parametro si noti che, come nel caso della velocità relativa, un inseguimento previsto di offset spaziale costante sia nella coordinata x che y, dà un valore nullo di errore in quanto, seppure su posizioni diverse target e soggetto ricoprono la stessa distanza, in termini di spazio percorso (Fig. 5.14, Fig.5.14).

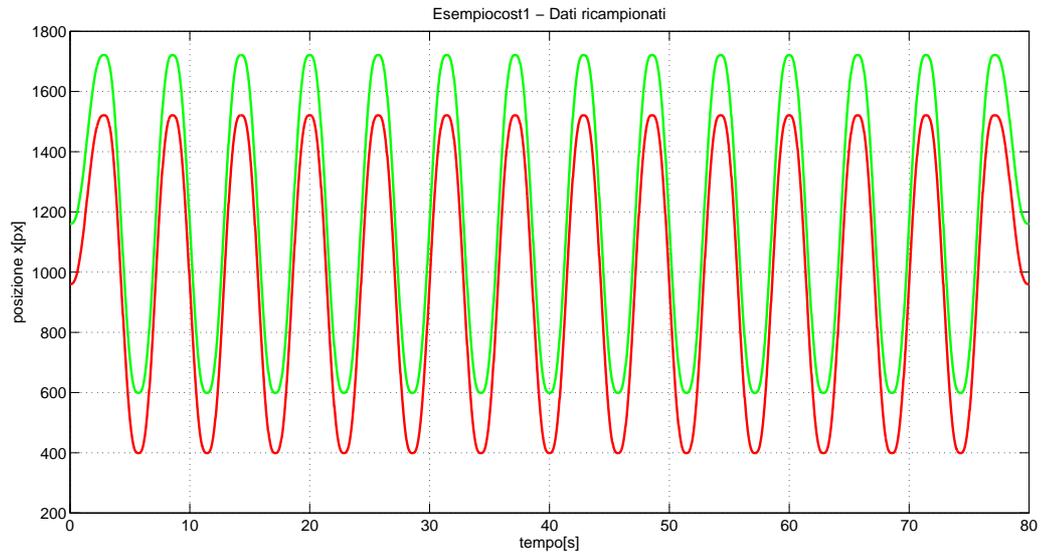


Figura 5.14: Tracciato di posizione di target (rosso) e soggetto(verde), di un inseguimento con offset spaziale in x.

Graficando i rispettivi profili di velocità si noterebbe che, salvo le imprecisioni legate al calcolo della derivata, essi coincidono.

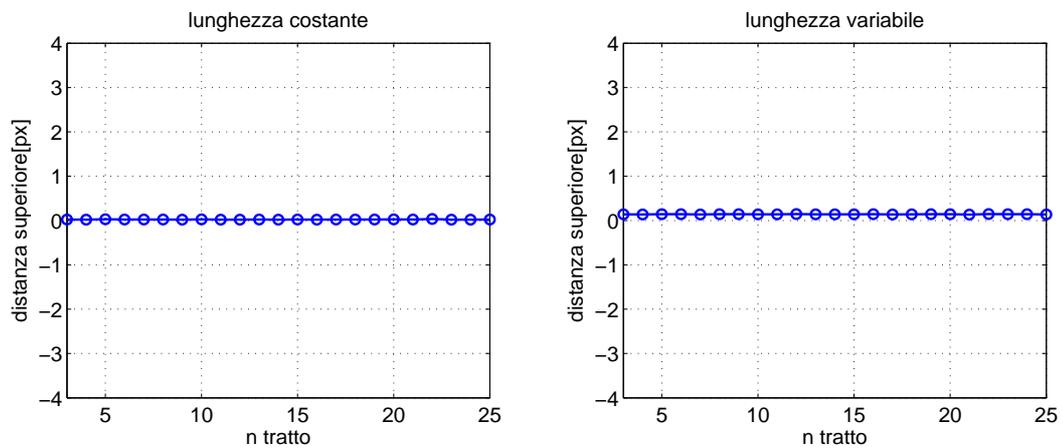


Figura 5.15: Integrale del modulo della velocità relativa nei tratti 3-25 per l'inseguimento riportato in Fig.

5.2.4 Errore di posizione medio (o ritardo medio)

Questo tipo di errore si basa sull'intuizione di pesare la differenza delle posizioni in x fra soggetto e target con la direzione del movimento del target, ovvero con il segno della velocità del target. Pertanto si può definire l'errore di posizione di ogni singolo campione come:

$$(x_s - x_t) * \text{sign}(v_{x,t})$$

dove :

- termini di segno negativo indicano configurazioni in cui, rispetto alla direzione della traiettoria da seguire, il soggetto si trova in *ritardo* rispetto al target (Fig.5.16)
- termini di segno positivo indicano configurazioni in cui, rispetto alla direzione della traiettoria da seguire, il soggetto si trova in *anticipo* rispetto al target (Fig. 5.17)

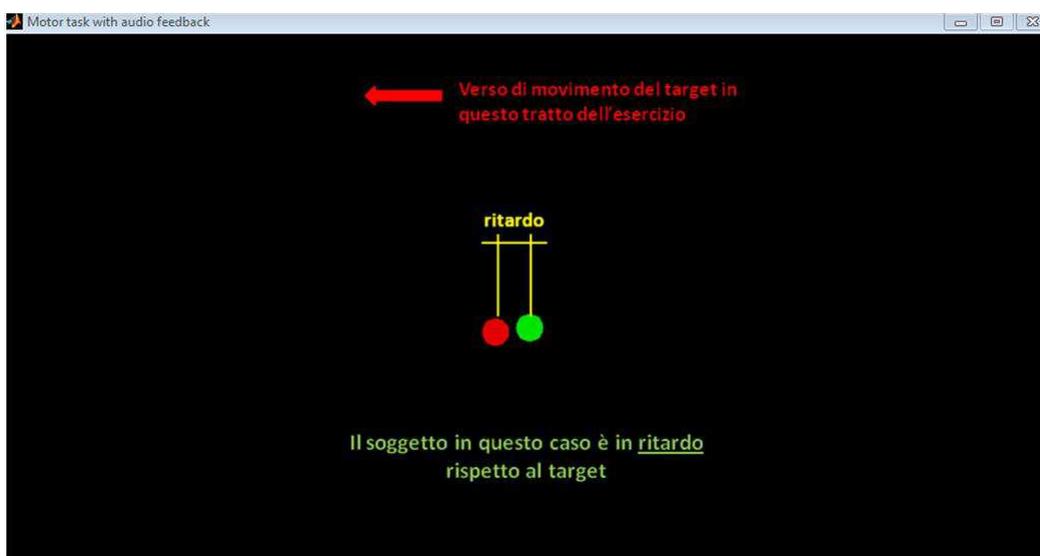


Figura 5.16: Situazione di ritardo nell'inseguimento.



Figura 5.17: Situazione di anticipo nell'inseguimento.

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

$$\varepsilon = \frac{1}{N} \sum_{k=1}^N \frac{(x_s(k) - x_t(k)) * \text{sign}(v_{x,t}(k))}{r}$$

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

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

Dall'osservazione della relazione che definisce questo parametro di errore si nota come contributi di errore uguali in modulo ma di segno opposto si annullano, mettendo in luce come l'inseguimento perfetto possa ottenersi anche quando si realizza un perfetto bilanciamento fra situazioni di anticipo e situazioni di ritardo durante tutta l'esecuzione del task. Numericamente ciò può essere provato dal calcolo di questo tipo di errore su di un task dove è stato costruito, ad esempio, un inseguimento a rotazione di raggio costante attorno al target (Fig. 5.18), o ad oscillazione a distanza costante dal target rispetto alla direzione x di movimento

(Fig. 5.19), ricordando che affinché ciò sia verificato, rotazioni e oscillazioni devono avvenire in un numero intero di volte per ogni tratto.

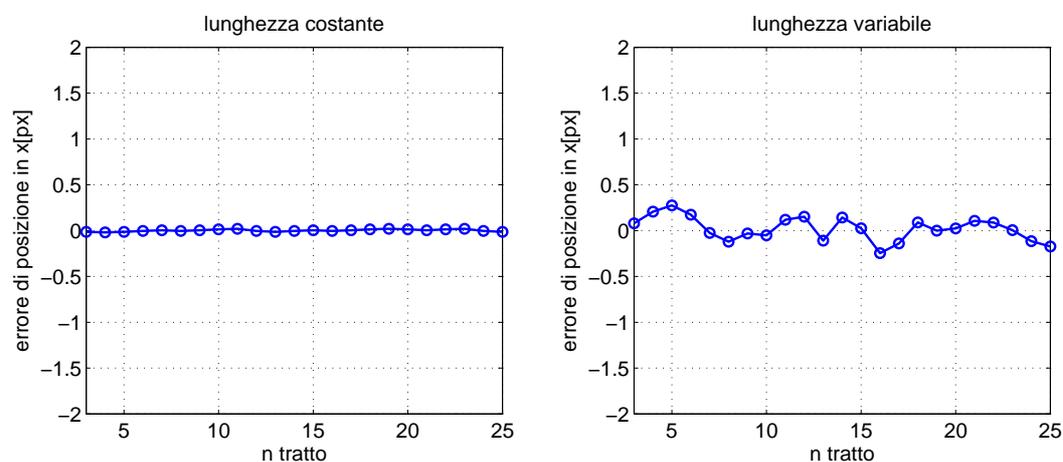


Figura 5.18: Errore di posizione medio nei tatti 3-25 per un task con inseguimento a rotazione attorno al target.

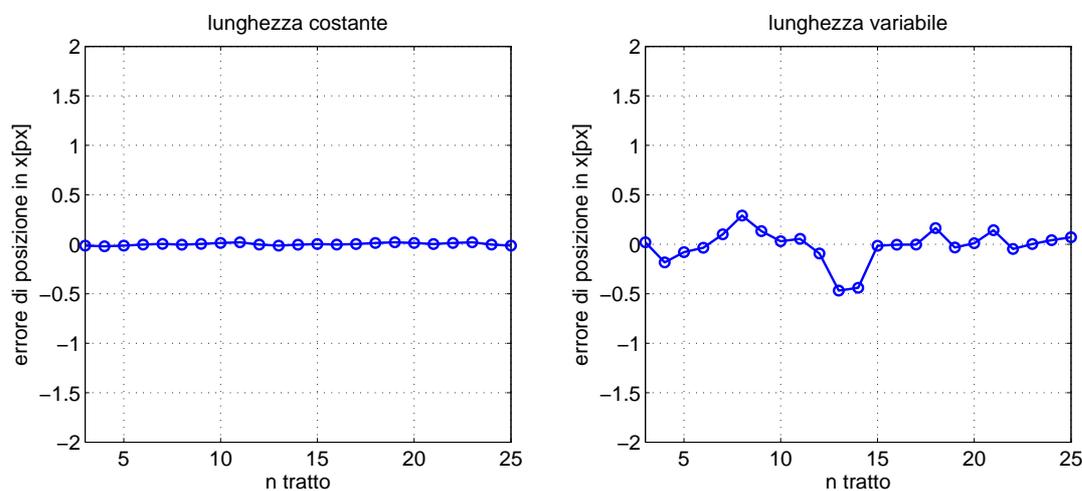


Figura 5.19: Errore di posizione medio nei tatti 3-25 per un task con inseguimento ad oscillazione in x rispetto al target.

Quando invece viene valutato su task con un inseguimento a ritardo costante e su task con inseguimento ad anticipo costante esso fornisce valori rispettivamente negativi e positivi come da definizione (Fig. 5.20 Fig. 5.21).

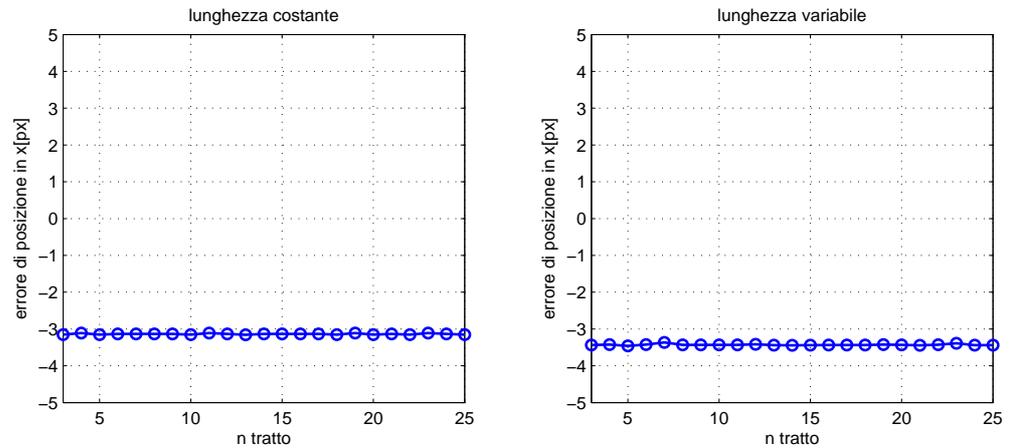


Figura 5.20: Errore di posizione medio nei tratti 3-25 di un task con inseguimento a ritardo costante.

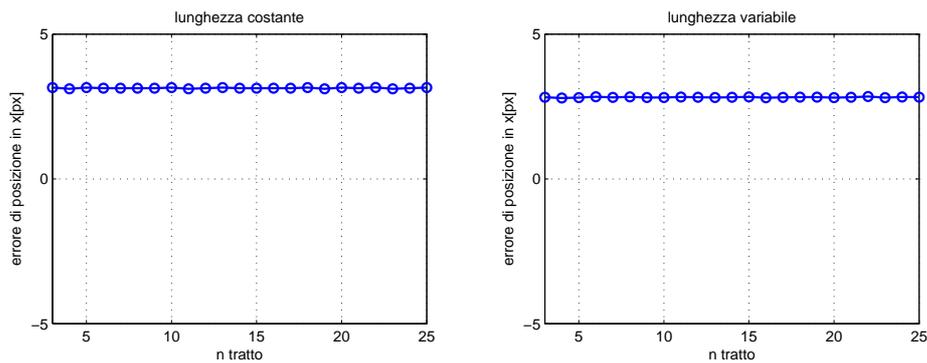


Figura 5.21: Errore di posizione medio nei tratti 3-25 di un task con inseguimento ad anticipo costante.

5.2.5 Lead error

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

$$\varepsilon = \frac{1}{N} \sum_{k=1}^N \frac{(x_s(k) - x_t(k)) * \text{sign}(v_{x,t}(k))}{r}$$

con $(x_s - x_t) * \text{sign}(v_{x,t}) > 0$.

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

Si noti come questo tipo di errore praticamente coincide, salvo brevi istanti di tempo in corrispondenza dei cambi di direzione, con quello globale di posizione nel caso di un inseguimento ad anticipo costante (Fig. 5.22), mentre risulta praticamente nullo in un inseguimento a ritardo costante (Fig.5.23). Il valore non nullo assunto dal parametro è dovuto ai cambi di direzione nei quali il soggetto viene a trovarsi, per brevi istanti di tempo, improvvisamente in ritardo rispetto al target.

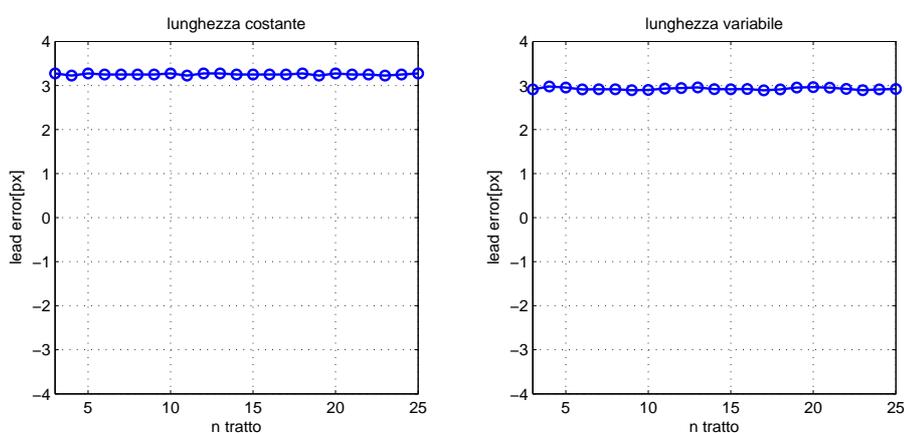


Figura 5.22: Lead error nei tratti 3-25 di un task con inseguimento ad anticipo costante.

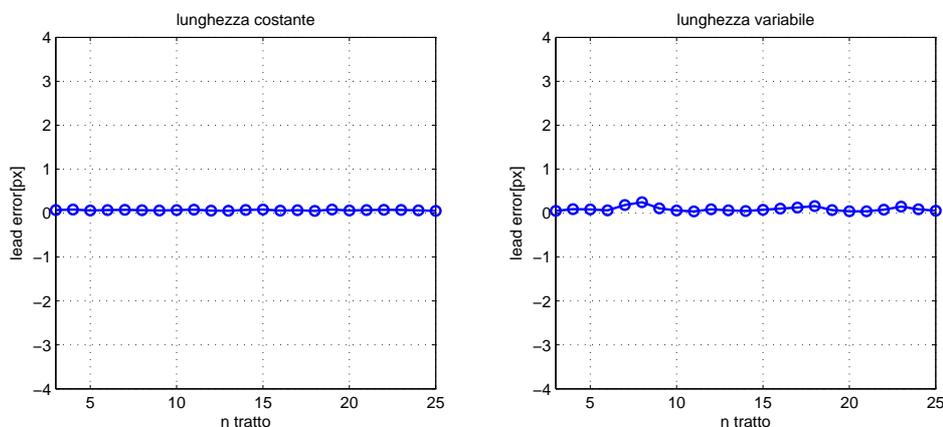


Figura 5.23: Lead error nei tratti 3-25 di un task con inseguimento a ritardo costante.

5.2.6 Lag error

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:

$$\varepsilon = \frac{1}{N} \sum_{k=1}^N \frac{(x_s(k) - x_t(k)) * \text{sign}(v_{x,t}(k))}{r}$$

con $(x_s - x_t) * \text{sign}(v_{x,t}) < 0$.

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

Si noti come questo tipo di errore praticamente coincide, salvo brevi istanti di tempo in corrispondenza dei cambi di direzione, con quello globale di posizione nel caso di un inseguimento a ritardo costante (Fig. 5.24), mentre risulta praticamente nullo in un inseguimento a anticipo costante (Fig.5.25). Il valore non nullo assunto dal parametro è dovuto ai cambi di direzione nei quali il soggetto viene a trovarsi, per brevi istanti di tempo, improvvisamente in anticipo rispetto al target.

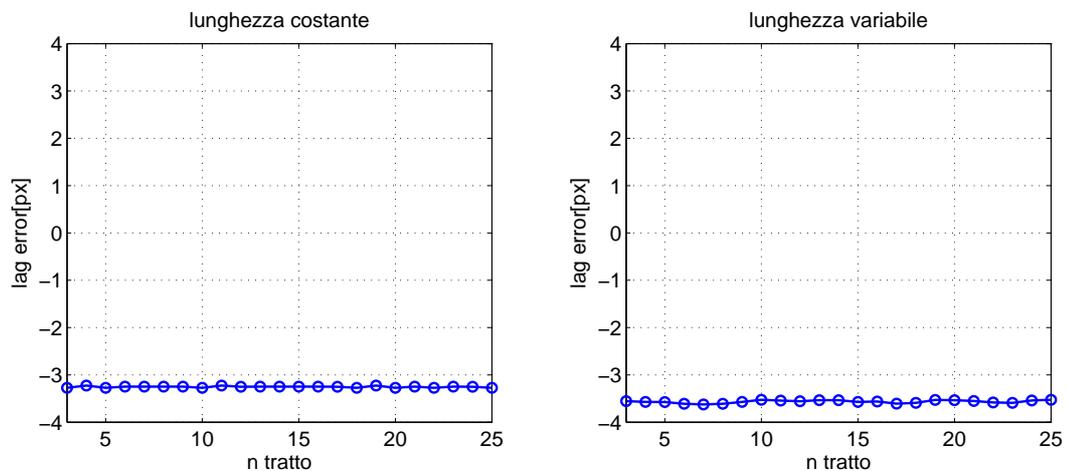


Figura 5.24: Lag error nei tratti 3-25 di un task con inseguimento a ritardo costante.

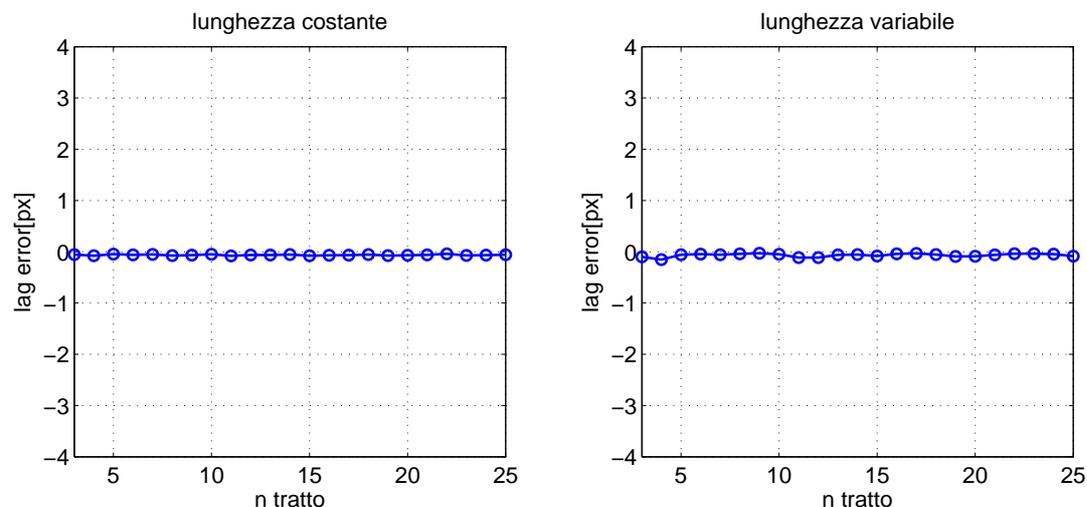


Figura 5.25: Lag error nei tratti 3-25 di un task con inseguimento ad anticipo costante.

5.3 Analisi statistica

5.3.1 Selezione delle impostazione per l'analisi

Per l'analisi statistica sui dati relativi ai test si è scelto di far rientrare nell'analisi tutti i soggetti componenti la popolazione testata, e di prendere in considerazione tutti e 6 i task eseguiti da ognuno di essi. Questa scelta è motivata soprattutto dalla richiesta di poter avere a disposizione la maggiore quantità di confronti possibili fra i vari task.

Di ogni task, per il calcolo degli errori si sono presi in considerazione solamente i dati relativi ai tratti dal quindicesimo al venticinquesimo, poichè si presume che un soggetto privo di disabilità motorie nella parte finale dell'esercizio abbia raggiunto il massimo livello di attenzione e abilità per garantire le massime prestazioni.

I calcoli dei vari parametri di errore sono stati eseguiti sulla versione della matrice ricampionata con vettore dei tempi dai campioni correttamente equispaziati nel tempo di 0.02s, e si è deciso di eliminare dall'analisi gli intervalli di tempo della durata di 2 secondi susseguenti ad un evento di irregolarità del timer (scatto).

5.3.2 Modalità di calcolo

Per il tipo di parametro di errore scelto si calcolano media e deviazione standard per ogni task. Nei dettagli:

- la media di un task per un dato parametro di errore è la media aritmetica delle medie relative ai singoli tratti
- la deviazione standard di un task per un dato parametro è la media aritmetica delle deviazioni standard relative ai singoli tratti

5.3.3 Test non parametrici

Per l'analisi statistica dei risultati forniti dai parametri di errore calcolati sui dati dei test, si è utilizzato un particolare strumento statistico di tipo non parametrico: *test di Wilcoxon per dati appaiati*.

Sono metodi statistici parametrici tutti i metodi 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.

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.

Dati che non si adattano alla distribuzione normale sono spesso quelli dei punteggi, votazioni, scores, utilizzati convenzionalmente da un osservatore (il medico, lo psicologo, l'insegnante, ecc.) per valutare fenomeni complessi quali l'intelligenza, la capacità di memoria, il rendimento a scuola, la produttività nel lavoro, la prestazione in un esercizio fisico, ecc. In tutti questi casi la scala non è riferita a grandezze fisiche, bensì a diversi livelli qualitativi di espressione del fenomeno.

Così come la media è un dato squisitamente parametrico, la mediana è l'analogo dato in campo non-parametrico. Mentre la media è scossa da valori molto piccoli o molto grandi, la mediana è del tutto stabile rispetto ai cosiddetti outliers. Pertanto si dovrebbe preferire la mediana (e i test non-parametrici) anche quando la distribuzione dei dati è normale ma si sospetta la presenza di dati spuri, con valori molto piccoli o molto grandi, dovuti a errori non rilevati nelle procedure sperimentali.

L'operazione iniziale e fondamentale della statistica non parametrica è quella di convertire i dati ponderali nel loro rango: brutto termine italiano che traduce l'inglese *rank*, che infatti significa anche posizione in graduatoria/classifica/ordine crescente. I ranghi siano molto robusti nei confronti di variazioni anche notevoli dei dati, inoltre anche nel caso in cui tutti i dati vengano trasformati in modo lineare (ad esempio additivo o moltiplicativo) o nonlineare (ad esempio esponenziale o logaritmico), i ranghi non cambiano affatto in quanto i dati mantengono la stessa posizione. In generale, qualsiasi trasformazione, purché monotonica, non altera i ranghi. Si evidenzia così come la statistica parametrica non sia influenzata dal tipo di distribuzione.

Vanaggi dei test non-parametrici I test non parametrici richiedono assunzioni meno stringenti sui dati rispetto a quelle necessarie per i consueti test parametrici, e questo soprattutto per piccoli campioni. Ad esempio il t-test per un campione richiede che i dati siano distribuiti secondo la distribuzione normale. Per il t-test per due campioni indipendenti si richiede inoltre che le deviazioni standard calcolate sui due campioni siano uguali. Se tali assunzioni o condizioni vengono violate, i risultati forniti dal test (come il P-value e gli intervalli di confidenza) possono risultare inattendibili.

Svantaggi dei test non parametrici Il maggior svantaggio delle procedure non parametriche è contenuto nel nome stesso: il termine "non-parametrico" indica che tali test non coinvolgono la descrizione e quindi la stima di alcun parametro. E' così difficile formulare ipotesi quantitative circa le reali differenze tra le popo-

lazioni. Ad esempio parlare di differenze tra i segni di due trattamenti non dice molto sulla reale differenza tra i trattamenti.

5.4 Test di Wilcoxon con GraphPad Prism

5.4.1 Software di analisi statistia utilizzato

Per le conclusioni statistiche sui dati memorizzati si è fatto ricorso al software specifico GraphPad Prism.

GraphPad Prism è un software statistico che combina grafica scientifica, statistiche e curve di interpolazione in un programma molto semplice da utilizzare. Prism ha tutte le capacità che ci si aspetterebbe da un programma scientifico di prima qualità grafica, ma ciò che lo rende veramente unico è il fatto che fornisce un'interfaccia utente-software veramente molto intuitiva e ricca di possibilità. 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. Lo stesso concetto vale anche per la creazione dei grafici.

Il procedimento di analisi si articola in due fasi:

- creazione del data-set: tabella contenente, per ogni task di ogni soggetto (collezionati per riga), il valore del parametro d'errore prescelto che riassume la performance dell'esecuzione e la deviazione standard relativa al calcolo di questo valore (Fig. 5.26)
- scelta di quale gruppo di dati analizzare e del tipo di test statistico da eseguire attraverso l'apposita finestra di dialogo (Fig.5.27)

Table format: Grouped		A		B		C		D		E		F	
		Task A		Task B		Task C		Task D		Task E		Task F	
		Mean	SD										
1	BENEDETTI GIULIO	0.883581	0.508671	1.091563	0.725283	0.916558	0.533270	1.011943	0.654802	1.119704	0.724487	1.088848	0.768275
2	CARAMMA LEA	1.028777	0.637357	1.240739	0.993692	1.123007	0.779648	0.998302	0.692021	1.208299	0.793938	1.387259	1.052197
3	CARDIN SILVIA	1.051206	0.660057	1.686266	1.173140	1.240092	0.675778	1.412016	1.012854	1.266252	0.653065	1.528968	1.065877
4	CARLI ANDREA	1.263232	0.866603	1.806064	1.669939	1.076044	0.747928	1.564437	1.148716	1.060079	0.683786	1.280561	0.803180
5	CARRARO MARTINA*	1.724549*	1.604082*	1.212178*	0.799320*	2.228731*	0.913078*	1.226668*	0.690566*	1.169831*	0.663774*	1.825395*	1.225645*
6	COSTANZI ENRICO	0.902098	0.582994	0.911766	0.687269	0.822447	0.512859	0.925529	0.652321	0.874578	0.596460	0.977973	0.664371
7	DE FAVERI VALENTINO	1.041902	0.703636	1.183481	0.827712	0.965127	0.638942	1.238423	0.813635	1.047460	0.676126	1.153692	0.774112
8	FRISON MARCO	1.036939	0.672278	0.975468	0.628431	1.069805	0.747059	0.979155	0.723050	1.002743	0.694222	1.016015	0.719340
9	GARBUJO ALESSIO	1.046620	0.680714	1.047585	0.616418	1.141456	0.678494	1.093642	0.647039	1.574494	0.900960	1.427806	0.968405
10	MARONI IDA	0.802585	0.542398	0.961085	0.695697	0.975197	0.606335	0.862945	0.652805	0.774997	0.497469	0.970132	0.680872
11	MILANI MATTEO	0.978998	0.732033	1.158631	0.800706	0.878770	0.653597	1.039013	0.675945	0.883669	0.592078	1.256656	0.840787
12	NARDO ALICE	1.009510	0.647371	1.203675	0.696408	0.929492	0.610471	1.449776	1.082960	0.669722	1.295006	0.774156	
13	NICOLI PAOLA	1.057881	0.732706	1.192325	0.807040	1.057077	0.797225	1.193484	0.799947	1.060462	0.791331	1.299701	0.853696
14	PASQUALIN ALICE *	0.000000*		1.122468*	0.805582*	0.925376*	0.610786*	1.323214*	0.926342*	0.878437*	0.546931*	1.055917*	0.727699*
15	PAVAN BEATRICE	1.121347	0.772186	1.072703	0.735195	0.876372	0.531659	1.818390	1.217252	1.127681	0.819255	1.171406	0.829627
16	ROMANI DANIELE	0.844431	0.592128	1.055504	0.823892	0.803190	0.541187	0.920319	0.610306	0.846615	0.516213	0.953685	0.636195
17	SARTORATO PAOLO	1.022625	0.724304	1.024490	0.728312	0.749074	0.477973	0.887566	0.600142	0.793023	0.498667	0.739634	0.464803
18	SMANIA GIOVANNI	1.150185	0.748841	1.189763	0.862043	1.036839	0.664774	1.325351	0.916865	0.918359	0.636135	1.319636	0.820198
19	ZAMPICINI ANDREA	0.844693	0.550691	1.018869	0.706468	0.911072	0.624157	0.999707	0.659882	0.868758	0.526555	1.115768	0.727351
20	ZANATA CHIARA	1.418375	0.871530	1.289556	0.883132	1.302391	0.777696	1.273698	0.720093	1.100892	0.592303	1.646327	1.042798

Figura 5.26: Esempio di set di dati.

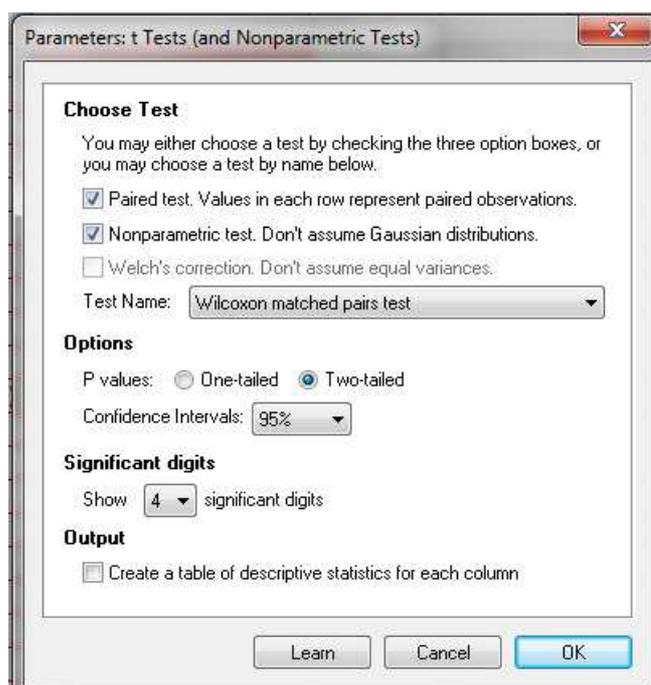


Figura 5.27: Esempio della finestra per la scelta del test.

5.4.2 Test di Wilcoxon per dati appaiati

Il test di Wilcoxon per il confronto tra due campioni appaiati è l'analogo non-parametrico del test t di Student per campioni appaiati. Questo test è molto utile

nel campo delle scienze comportamentali e consente di determinare la direzione e la grandezza della differenza dei dati appaiati.

Più precisamente ci si riferisce ad esso come test di Wilcoxon dei ranghi con segno per dati appaiati (dall'inglese "Wilcoxon matched pairs signed ranks test"). Il test di Wilcoxon analizza, per ogni soggetto, solo le differenze fra misurazioni accoppiate.

Interpretazione del P-value Il valore del parametro P che il test restituisce risponde alla domanda: nel caso in cui la mediana delle differenze è zero globalmente, qual è la possibilità che un campione casuale dell'intera popolazione risulti in una mediana delle differenze distante dallo zero almeno quanto osservato in questo esperimento?

Se il valore di P è piccolo, si può rigettare l'idea che la differenza considerata è una coincidenza, e concludere che invece che la popolazione ha mediane differenti.

Se il valore di P è elevato, allora i dati considerati non danno alcun motivo per concludere che le mediane globali differiscono. Ciò non significa che esse sono le stesse, quanto piuttosto che non si hanno a disposizione sufficienti prove per dimostrare che esse sono diverse.

Indipendenza delle coppie Il risultato del test di Wilcoxon ha senso solo quando gli accoppiamenti sono indipendenti, qualsiasi fattore che causi una differenza fra valori accoppiati deve influenzare solo quella coppia.

Efficacia dell'accoppiamento Il punto focale nell'utilizzo dei test con accoppiamento è di controllare la variabilità di un esperimento. L'appaiamento dei dati nell'analisi permette di eliminare, attraverso la differenza, cause di errore non controllate che influiscono in egual modo sulle misurazioni, tuttavia l'accoppiamento deve essere parte intrinseca della filosofia dell'esperimento eseguito e non qualcosa di inserito dopo la memorizzazione dei dati.

Il software esegue un test sull'efficacia dell'accoppiamento calcolando il coefficiente di Spearman e un corrispondente valore di P. Se il coefficiente è positivo e P piccolo, i due gruppi sono significativamente correlati, ciò giustifica l'utilizzo del test per dati appaiati.

Test a una o due code Si utilizza il test a due code quando non si è certi della direzione della differenza fra le mediane dei gruppi. Se si sceglie il test con P-value ad una coda, si è fatta l'implicita assunzione che si è già a conoscenza di quale gruppo presenterà la mediana più larga ancor prima di raccogliere i dati.

Capitolo 6

Interpretazione dei risultati

Per chiarezza si è scelto di separare la descrizione dei problemi di elaborazione ed analisi dei dati dalla presentazione dei risultati finali da questi ottenuti. Ciò non toglie che questo capitolo rappresenti la naturale prosecuzione del capitolo precedente, al quale si rinvia per una comprensione più generale del quadro finale delineato. Vengono infatti presentati, e si cerca di darne una concreta interpretazione, i risultati relativi ai test sui soggetti, ottenuti dall'analisi statistica eseguita con i parametri le modalità ampiamente descritte nel capitolo precedente. Con riferimento al test di Wilcoxon si ha significatività tra le differenze per valori di P inferiori a 0.05

6.1 Distanza media

Di seguito sono riportati i risultati relativi all'analisi del parametro distanza media. Si ricordi che in giallo sono riportate le differenze di segno negativo, in verde quelle di segno positivo.

differenze	significatività	valore di p
A-B	si	0.0031
C-D	si	0.0383
E-F	si	0.0003
A-C	no	0.7937
B-D	no	0.0967
A-E	no	0.7022
B-F	no	0.9256
C-E	no	0.7795
D-F	no	0.0826
A-D	si	0.0313

Figura 6.1: Tabella riassuntiva dei risultati del parametro distanza.

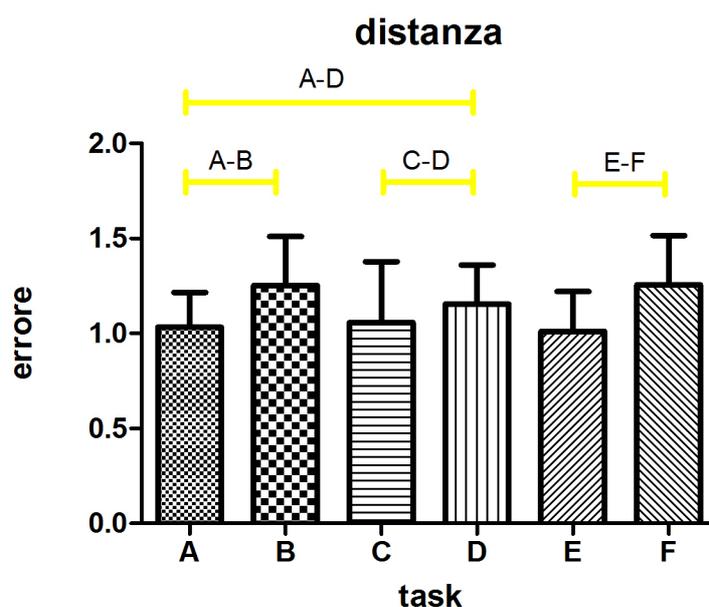


Figura 6.2: Grafici dell'analisi relativa alla distanza.

Si può affermare che per ognuna delle tre modalità audio testate, l'esercizio con traiettoria a tratti di lunghezza fissa risulta sempre più facile del corrispondente esercizio con traiettoria a lunghezza variabile, in termini di distanza media dal target durante l'inseguimento. Quando la traiettoria è difficilmente prevedibile il soggetto porta a termine l'inseguimento compiendo dei movimenti attorno al target di ampiezza maggiore, indipendentemente dal tipo di feedback audio forn-

to. Al contrario non vi sono variazioni significative quando, nella stessa modalità di modalità di esercizio, si cambia la tipologia di feedback uditivo.

Altro risultato interessante riguarda il confronto tra il task A (lunghezza costante, no audio) e il task D (lunghezza variabile, audio sulla velocità del target) che risulta significativo, con segno della differenza negativo, a testimoniare come un esercizio a tratti variabili rimanga più difficoltoso per quanto riguarda per l'inseguimento di un esercizio a tratti costanti, nonostante l'esecuzione del primo venga agevolata in termini di distanza media dal target dall'aggiunta dei feedback uditivo (risulta significativo, con lo stesso segno il confronto A-F).

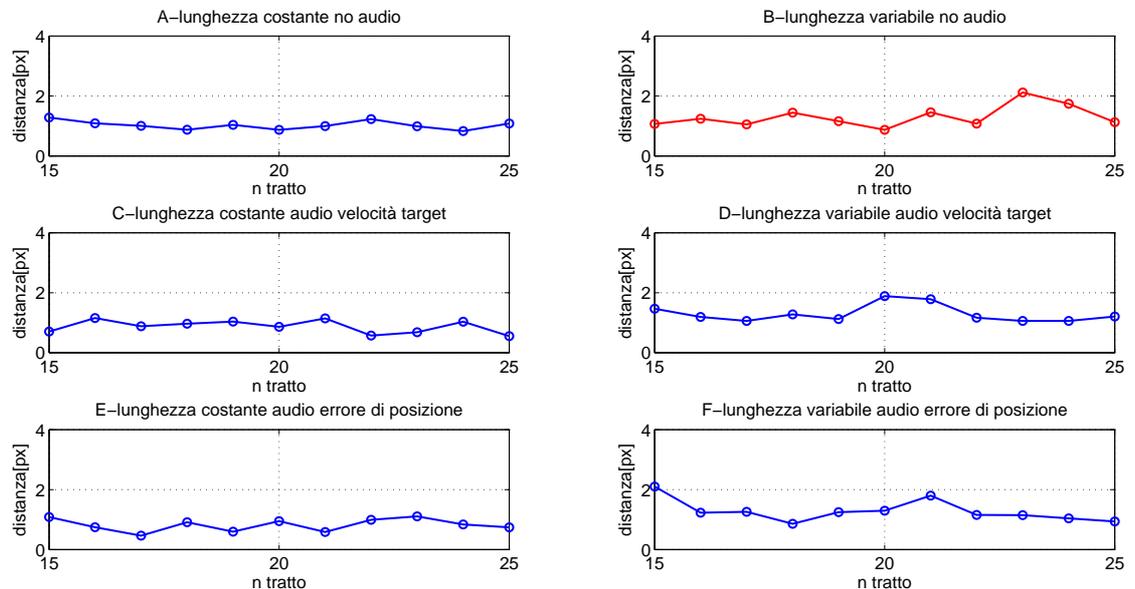


Figura 6.3: Grafici dell'andamento dell'errore di distanza. Nella colonna di sinistra i grafici hanno un errore mediamente più basso.

6.2 Media del modulo della velocità relativa

Di seguito sono riportati i risultati relativi all'analisi del parametro integrale del modulo della velocità relativa che fornisce informazioni connesse allo spazio percorso inutilmente. Si ricordi che in giallo sono riportate le differenze di segno negativo, in verde quelle di segno positivo.

differenze	significatività	valore di p
A-B	si	0.0002
C-D	si	<0.0001
E-F	si	0.0001
A-C	no	0.5328
B-D	no	0.4666
A-E	no	0.9839
B-F	no	0.3803
C-E	no	0.9533
D-F	no	0.0500
A-D	si	0.0005

Figura 6.4: Tabella riassuntiva dell'analisi sul modulo della velocità relativa medio.

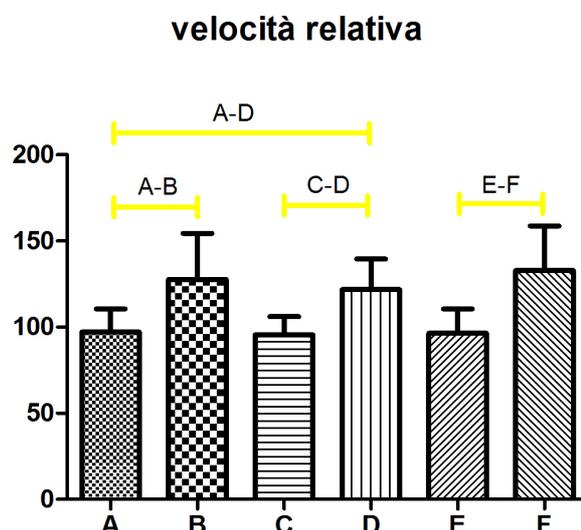


Figura 6.5: Grafico dell'analisi relativa alla modulo della velocità relativa medio.

I risultati che fornisce il modulo medio della velocità relativa sono analoghi a quelli forniti dal parametro d'errore distanza media solo che, confrontando i valori assunti dal parametro P, si può notare che essi permettono un'analisi ancor più significativa dati i bassissimi valori di P.

Riassumendo quindi, anche in relazione all'entità del modulo della velocità relativa la presenza del feedback audio mantiene l'esercizio con tratti a lunghezza variabile più difficile di quello con tratti a lunghezza fissa, e non introduce significative variazioni all'interno della stessa tipologia di esercizio.

6.3 Integrale del modulo della velocità relativa

Di seguito sono riportati i risultati relativi all'analisi del parametro modulo della velocità relativa medio. Si ricordi che in giallo sono riportate le differenze di segno negativo, in verde quelle di segno positivo.

differenze	significatività	valore di p
A-B	si	0.0001
C-D	si	>0.0001
E-F	si	>0.0001
A-C	no	0.6435
B-D	no	0.9851
A-E	no	0.9839
B-F	no	0.2708
C-E	no	0.8666
D-F	no	0.0967
A-D	si	0.0002

Figura 6.6: Tabella riassuntiva dell'analisi eseguita sull'integrale del modulo della velocità relativa.

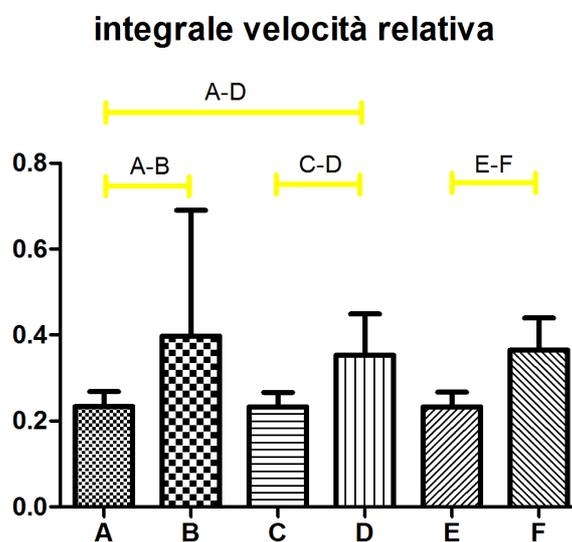


Figura 6.7: Grafici dell'analisi relativa all'integrale del modulo della velocità relativa.

Le considerazioni sono del tutto analoghe a quelle fatte per i due precedenti parametri di errore.

In questo caso però si intravede un primo effetto imputabile alla presenza del feedback uditivo durante l'esecuzione di un esercizio. Infatti la grande variabilità di prestazione tra i soggetti nell'esecuzione del task B (indicata dalla elevata lunghezza del trattino che sovrasta la corrispondente colonna), viene notevolmente ridotta negli equivalenti task dotati di feedback audio, cioè i task D e F. Ciò significa che, per questo parametro di analisi, l'influenza del feedback audio è quella di ricompattare la popolazione, in termini di similarità delle prestazioni, nell'esecuzione di task complicati.

6.4 Errore di posizione medio

Di seguito sono riportati i risultati relativi all'analisi del parametro errore di posizione medio. Si ricordi che in giallo sono riportate le differenze di segno negativo, in verde quelle di segno positivo.

differenze	significatività	valore di p
A-B	no	0.8248
C-D	no	0.2549
E-F	no	0.7150
A-C	si	0.0016
B-D	si	0.0196
A-E	no	0.4566
B-F	no	0.8666
C-E	si	0.0012
D-F	si	0.016
A-D	si	0.0230

Figura 6.8: Tabella riassuntiva dell'analisi sull'errore medio di posizione.

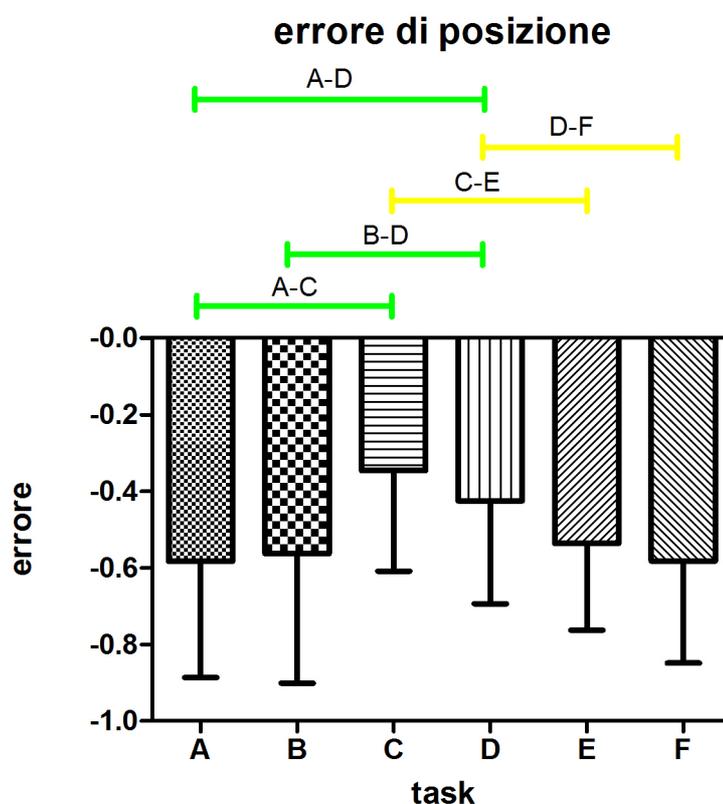


Figura 6.9: Grafico dell'analisi relativa all'errore di posizione medio.

Per questo parametro d'errore si nota che all'interno della stessa modalità di audio (A-B, C-D, E-F) non si ha significatività nel confronto fra traiettoria a tratti costanti e traiettoria a tratti variabili. Ciò significa che la differenza fra il ritardo medio che caratterizza l'esercizio facile e quello che caratterizza l'esercizio difficile non è statisticamente differente.

Se si analizza come evolve l'errore a seconda del feedback utilizzato nei task a traiettoria costante si ottiene:

- la presenza nell'esercizio del feedback legato alla velocità del target riduce, con significatività, il ritardo medio rispetto a quello eseguito in assenza di audio
- la presenza nell'esercizio del feedback audio legato alla velocità del target riduce, con significatività, il ritardo medio dell'esercizio rispetto a quello

eseguito con feedback sull'errore di posizione

- la presenza nell'esercizio del feedback audio legato all'errore di posizione non dà significative differenze nel ritardo medio rispetto all'esercizio eseguito in assenza di feedback audio

L'analisi dell'evoluzione dell'errore a seconda del feedback utilizzato nei task a traiettoria dai tratti variabile offre le stesse conclusioni.

Si è dunque ottenuto un importante risultato: l'aggiunta del feedback audio sulla velocità del target diminuisce il ritardo medio nell'esecuzione di uno stesso (o comparabile) task (Fig.6.10)

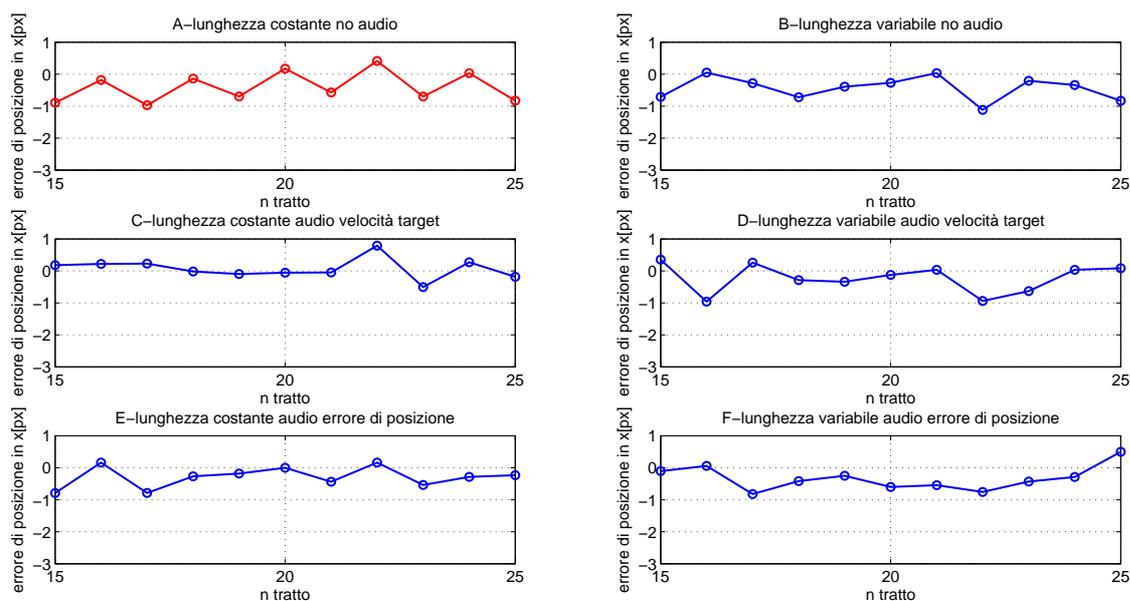


Figura 6.10: Grafici dell'andamento del ritardo medio. Gli andamenti della seconda riga sono mediamente più alti rispetto a quelli della prima riga, e della terza riga, ad indicare che l'aggiunta del feedback audio legato alla velocità del target riduce il ritardo medio rispetto alle alte tipologie di audio.(task .)

Un'ulteriore analisi relativa a questo tipo di errore riguarda la diminuzione, significativa, del ritardo nel passaggio da un esercizio a traiettoria facile (lunghezza dei tratti costante) privo di feedback audio ad un esercizio a traietto-

ria difficile (lunghezza dei tratti variabile) con feedback audio sulla velocità del target.

6.5 Lead error

Di seguito sono riportati i risultati relativi all'analisi del parametro lead error medio. Si ricordi che in giallo sono riportate le differenze di segno negativo, in verde quelle di segno positivo.

differenze	significatività	valore di p
A-B	si	0.0006
C-D	si	0.0318
E-F	si	0.0006
A-C	si	0.0040
B-D	no	0.6677
A-E	no	0.0872
B-F	no	0.7795
C-E	si	0.0217
D-F	no	0.7510
A-D	si	0.0008

Figura 6.11: Tabella riassuntiva dell'analisi sul lead error.

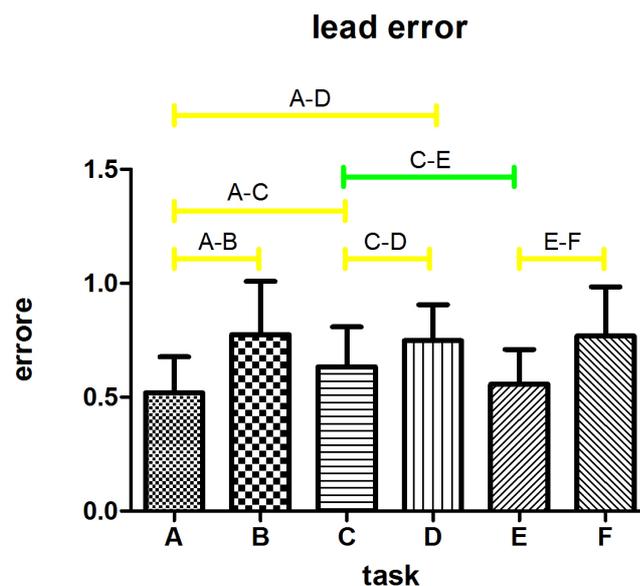


Figura 6.12: Grafici dell'analisi relativa al lead error.

Per quanto riguarda l'evoluzione del lead error per la stessa modalità di audio si denota un aumento, statisticamente significativo, negli esercizi con traiettoria con tratti di lunghezza variabile. Ciò può essere spiegato considerando l'imprevedibilità dei cambi di direzione che determinano improvvisi ribaltamenti da situazioni di ritardo a situazioni di anticipo e che, probabilmente, spingono il soggetto a ad anticipare il target per non farsi "sorprendere" dal cambio di direzione.

Lo stesso effetto è provocato dalla presenza del feedback audio legato alla velocità del target nei task a lunghezza dei tratti costante, rispetto ai relativi task privi di audio o con audio legato all'errore di posizione. La conoscenza, per via uditiva oltre che visiva della velocità del target, facilita la previsione del punto dove il target invertirà il senso di marcia, e stimola il soggetto ad anticipare questo evento.

Queste considerazioni, combinate assieme, giustificano come risultati significativo, il confronto del lead error fra il task D e il task A, con un anticipo mediamente superiore nel task D.

6.6 Lag error

Di seguito sono riportati i risultati relativi all'analisi del parametro lag error medio. Si ricordi che in giallo sono riportate le differenze di segno negativo, in verde quelle di segno positivo.

differenze	significatività	valore di p
A-B	si	0.0084
C-D	si	0.0178
E-F	si	0.0006
A-C	si	0.0283
B-D	si	0.0085
A-E	no	0.5596
B-F	no	0.9851
C-E	no	0.3046
D-F	si	0.0239
A-D	no	0.4094

Figura 6.13: Tabella riassuntiva dell'analisi sul lag error.

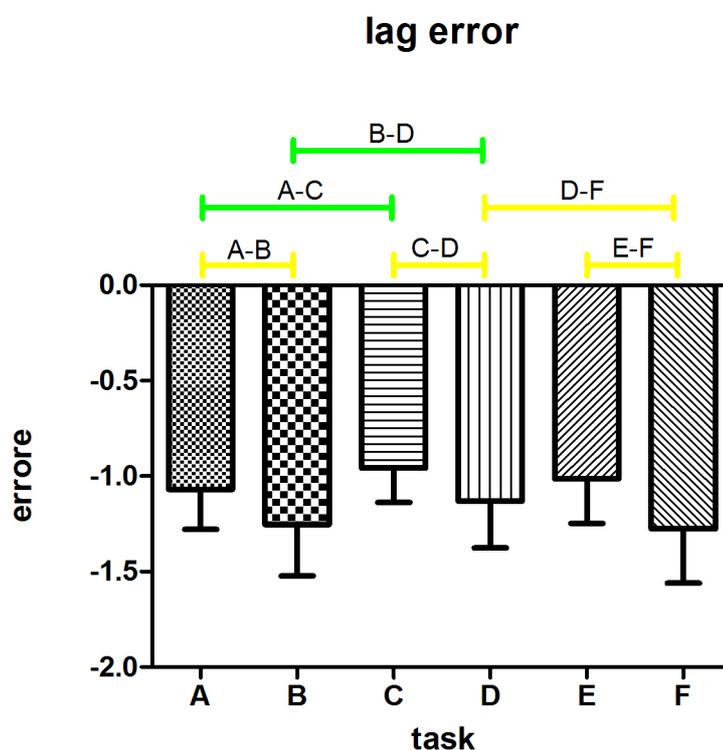


Figura 6.14: Grafici dell'analisi relativa al lag error.

Per quanto riguarda l'evoluzione del lag error in esercizi con la stessa modalità di audio si denota un aumento, statisticamente significativo, in quelli a traiettoria dai tratti di lunghezza variabile rispetto al corrispondente a traiettoria dai tratti di lunghezza fissa. L'aumento del peso delle componenti di ritardo è dovuto essenzialmente alla maggior complessità dell'esercizio.

Per questo tipo di parametro d'errore si verificano anche interessanti andamenti per quanto riguarda uno stesso esercizio proposto però con tipi di feedback audio differenti. Si vede infatti (Fig.6.14) che il lag diminuisce, con significatività, passando da un esercizio privo di audio ad un esercizio con audio legato alla velocità del target, indipendentemente dal tipo di traiettoria proposta. E' questo un risultato importantissimo a favore della ipotesi formulata all'inizio della trattazione riguardo l'utilità di un appropriato feedback uditivo nell'esecuzione di task motori.

Un'ulteriore considerazione sui risultati forniti da questo tipo di errore riguarda il confronto fra il task A (facile ma privo di audio) e il task D (difficile ma con

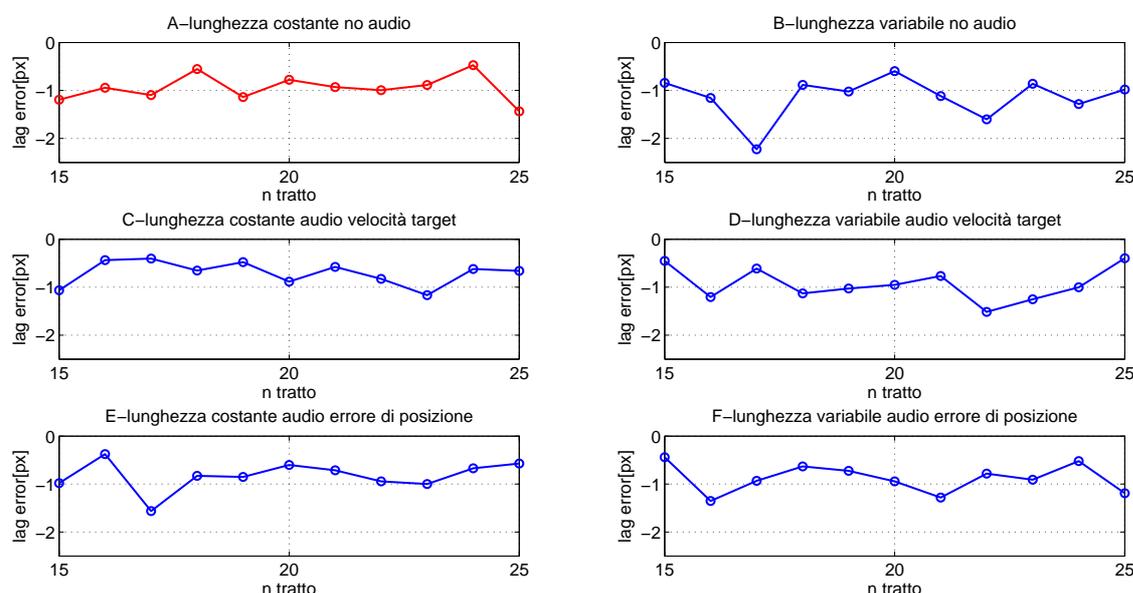


Figura 6.15: Grafici dell’andamento del lag error nei vari tratti. Si vede come i grafici della seconda riga siano posizionati mediamente più vicini allo zero rispetto ai corrispondenti della prima riga.

audio). Si può constatare infatti che, il task B risulti mediamente più complicato nell’inseguimento rispetto al task A. L’aggiunta del feedback sulla velocità del target contribuisce ad abbassare il livello di difficoltà dell’esercizio a traiettoria variabile, risulta infatti significativo il confronto B-D, con D “inferiore” rispetto a B. Ma si può dire ancor di più, in quanto l’aggiunta del feedback appropriato rende l’esercizio con traiettoria difficile (task D) non significativamente differente dall’esercizio iniziale, quello dalla traiettoria facile ma privo di audio (task A), come a dimostrare la capacità di un buon feedback audio a controbilanciare l’incremento di difficoltà introdotto dalla variazione di traiettoria.

6.7 Riassunto dei risultati

Tutte i risultati precedentemente esposti portano a queste conclusioni (Fig., 6.16):

- Qualsiasi sia il tipo di feedback uditivo utilizzato nell’esercizio, il task con

traiettoria dai tratti variabili comporta una minore accuratezza nell'inseguimento rispetto al task con traiettoria dai tratti fissi. In questa tipologia di task si constata una maggiore ampiezza dei movimenti correttivi, che avvengono in maniera brusca con enormi differenze di velocità (analisi di distanza e di velocità relativa). Il ritardo medio rimane pressochè invariato in quanto risultano aumentate in maniera simile sia la componente di lead che di lag error.

- L'audio sulla velocità del target rispetto all'audio assente ha l'effetto di ridurre il ritardo medio, fatto dovuto alla contemporanea riduzione del lag error e aumento del lead error. Il fatto che non si verifichi una riduzione delle ampiezze dei movimenti correttivi attorno al target (l'osservazione dei grafici del task a lunghezza variabile suggerisce il contrario, ma l'impressione visiva non è confermata dall'analisi statistica), porta a concludere che il feedback uditivo sulla velocità del target ha come principale effetto quello di traslare nella direzione di anticipo il range di variabilità dei movimenti di correzione del soggetto sulla posizione del target durante l'inseguimento.
- L'audio legato all'errore di posizione non dà significativi risultati rispetto alle performance in condizioni di assenza di audio
- L'audio sulla velocità del target rispetto a quello sull'errore di posizione ha sulle performance gli stessi effetti riscontrati nel confronto con audio nullo: riduzione del ritardo medio dovuto alla traslazione del range di variabilità dei movimenti di correzione. Inoltre, seppur non sia abbastanza marcato da risultare significativo dal punto di vista statistico, si mette in luce il fatto di come la sensibile riduzione del lag comporti una riduzione nella ampiezza del range di variabilità degli errori nel task a traiettoria variabile.

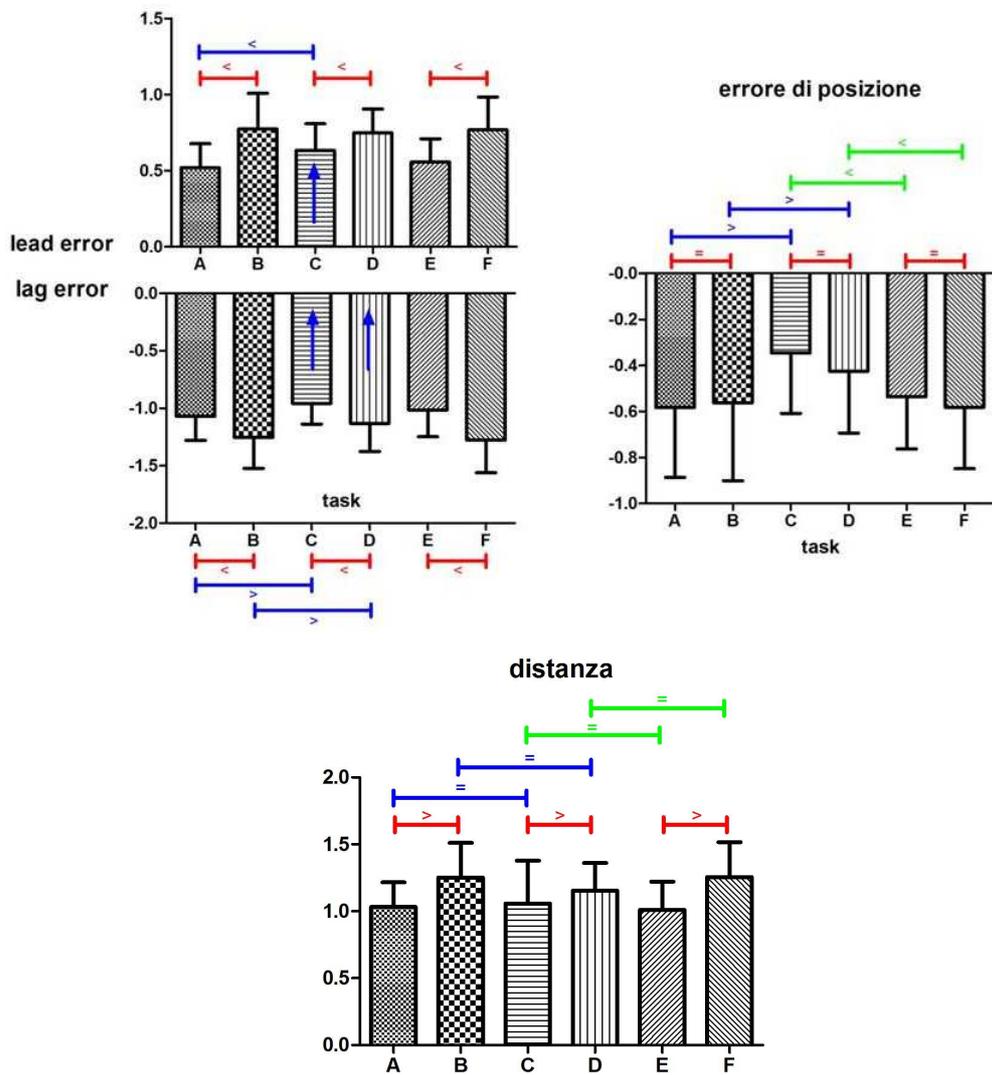


Figura 6.16: Riassunto dei risultati.

Conclusioni

I risultati ottenuti sono di buon auspicio per un utilizzo efficace del feedback audio in sedute di riabilitazione. Nonostante si siano effettuati i test su dei soggetti sani si è riuscito a riscontrare la presenza di qualche interessante aspetto connesso alla presenza del feedback audio durante l'esecuzione dei task. Come i risultati dimostrano, comunque non è sufficiente introdurre un qualunque tipo di feedback per ottenere dei miglioramenti. Il feedback infatti deve essere accuratamente progettato, analizzando quali sono i parametri che al un soggetto può far comodo apprendere per via immediata, come quella uditiva, al fine di accurare l'inseguimento del target. Nello studio descritto si è visto come mentre il feedback relativo alla velocità del target, informazione non valutabile precisamente attraverso il solo canale visivo, si configuri come un elemento utile nell'esecuzione del task, quello legato all'errore di posizione, informazione del resto già completamente acquisita tramite feedback visivo, non offra alcun risultato significativo.

Interessante sarebbe poter progettare e valutare nuove tipologie di feedback, legate ad altre grandezze o addirittura ad una combinazione di queste, così come si potrebbero individuare nuovi parametri d'errore per descrivere analiticamente il livello di prestazione raggiunto nel task.

Infine si potrebbero proporre nuovi tipi di traiettorie, magari più complesse, le quali possono risultare molto probabilmente più adeguate per trarre considerazioni sul feedback uditivo quando si tratta con soggetti sani.

Bibliografia

- [1] T. I. L. S. W. Group, “Prevalence of chronic diseases in older italians: comparing self reported and clinical diagnoses,” *Australian Journal of Physiotherapy*, vol. 1002, pp. 26–995, 1997.
- [2] A. Caperna, “Ictus, percorsi assistenziali contro la malattia, la proposta del ministro fazio,” *Salute Domani*, 2010.
- [3] C. A, M. C, and D. N. M, “High stroke incidence in the prospective community-based,” *Aquila registry 1994-1998*, vol. Stroke 1997, no. 6, pp. 28–2500, 1997.
- [4] D. Carlo, A. Baldereschi, M. G. C, and al, “Stroke in elderly population: incidence and impact on survival and daily function,” *The Italian Longitudinal Study on Aging*, vol. Cerebrovascular Disease, no. 50, pp. 16–141, 2003.
- [5] R. Colombo, F. Pisano, S. Micera, A. Mazzone, C. Delconte, M. C. Carrozza, P. Dario, and G. Minuccio, “Robotic techniques for upper limb evaluation and rehabilitation of stroke patients,” *IEEE Trans Neural Syst Rehabil Eng*, vol. 13, no. 3, pp. 313–324, 2005.
- [6] E. Carraro, S. Masiero, G. Rosati, and C. Ferraro, “Recupero della forza muscolare dopo precoce mobilizzazione passiva nel paziente emiplegico,” *EUROPA MEDICOPHYSICA*, vol. 44, no. 3, pp. 1–3, 2008.
- [7] A. Roda, F. Avanzini, S. Masiero, , and G. Rosati, “Auditory feedback in robot-assisted neuro-rehabilitation: state of the art and future prospects,” -, vol. -, no. -, pp. -, 2010.

-
- [8] M. H. Thaut, G. C. MacIntosh, and R. R. Rice, "Rhythmic facilitation of gait training in hemiparetic stroke rehabilitation," *Clin rehabil*, vol. 17, no. -, pp. 713–722, 2003.
- [9] S. Koelsch, E. Kasper, D. Sammler, K. Schulze, T. G. T, and A. D. Friederici, "Music, language and meaning: brain signatures of semantic processing," *Nat Neurosci*, vol. 7, pp. 302–307, 2004.
- [10] K. E. Watkins, A. P. Strafella, and T. Paus, "Seeing and hearing speech excites the motor system involved in speech production," *Neuropsychologia*, vol. 41, no. 9, pp. 989–994, 2003.
- [11] R. A. Magill, "Motor learning and control: Concepts and applications," *New York: McGraw-Hill*, no. 7th Edition, 2003.
- [12] P. M. V. Vliet and G. Wulf, "Extrinsic feedback for motor learning after stroke: What is the evidence," *Disability and Rehabilitation*, vol. 28, no. -, pp. 831–840, 2006.
- [13] R. Secoli, G. Rosati, and J. Reinkensmeyer, "Using sound feedback to counteract visual distractor during robot-assisted movement training," -, vol. -, no. -, pp. -, 2009.