Università degli studi di Pisa



Dipartimento di Ingegneria dell'informazione

Corso di Laurea Specialistica in Ingegneria dell'Automazione

Tesi di Laurea Specialistica

Studio e validazione sperimentale del controllo con reti neurali di esoscheletri robotici mediante sEMG

Candidato: Stefano Dettori Relatori: Prof. Carlo Alberto Avizzano Prof. Antonio Frisoli Dott. Ing. Claudio Loconsole

> Sessione di Laurea del 13/12/2013 Anno Accademico 2012/2013

Università degli studi di Pisa



Dipartimento di Ingegneria dell'informazione

Corso di Laurea Specialistica in Ingegneria dell'Automazione

Tesi di Laurea Specialistica

Studio e validazione sperimentale del controllo con reti neurali di esoscheletri robotici mediante sEMG

Candidato: Stefano Dettori Relatori: Prof. Carlo Alberto Avizzano Prof. Antonio Frisoli Dott. Ing. Claudio Loconsole

> Sessione di Laurea del 13/12/2013 Anno Accademico 2012/2013

 $\begin{array}{c} Alla \ mia \ famiglia, \\ A \ Sebastiano \end{array}$

Sommario

Il lavoro presentato in questa tesi è composto principalmente da due attività. Nella prima è stato proposto e validato un metodo per la stima delle coppie ai giunti di spalla e gomito basata su segnali elettromiografici di superficie (sEMG) per forze esercitate lungo il piano sagittale in condizioni isometriche. Nella seconda attività, il sistema di stima delle coppie ai giunti è stato integrato all'interno di un controllo per l'esoscheletro leggero L-Exos ed è stato validato. Dopo una breve introduzione sulle motivazioni del lavoro, son stati esposti lo stato dell'arte del controllo elettromiografico e dei sistemi robotici controllati tramite elettromiografia e le tecniche elettromiografiche. Viene descritto il sistema esoscheletrico L-Exos, il sistema di acquisizione e le metodologie di processamento dei segnali sEMG. Viene descritto il sistema di stima delle coppie ai giunti basato sulle reti neurali Time delay Neural Network (TDNN) ed i segnali sEMG, focalizzando l'attenzione sull'acquisizione dei dataset ed i metodi utili all'addestramento, la validazione ed il test delle reti. Viene presentato lo studio di ottimizzazione della struttura delle TDNN ed i risultati ottenuti sulla generalizzazione. Infine viene descritto il sistema per controllo on-line in posizione dell'esoscheletro L-Exos, implementato a partire dal sistema di stima delle coppie sviluppato nell'attività precedente e presentato il relativo studio sperimentale di validazione.

Abstract

The work presented in this thesis consists primarily of two activity. In the first one, a superficial electromyographic signal (sEMG) based method for the prediction of elbow and shoulder joint torque in isometric condition was proposed and validated. In the second, the joints torque prediction system was integrated in the Light exoskeleton (L-Exos) control system and next validated. After an essential introduction, EMG control systems and EMG controlled exoskeleton state of art and electromyographic techniques were described. L-Exos system, acquisition system and sEMG signal processing methods were described. The proposed joints torques prediction method, by using Time Delaly Neural Network (TDNN) and sEMG signals, was also illustrated, focusing in dataset acquisition and methods for networks training, validation and test. In the next section, an TDNN architecture optimization study was introduced and results were shown. Finally L-Exos on-line position control, implemented with torque prediction system developed in the previous activity, was described and validated with experimental study.

Indice

1	Intr	oduzio	one	11							
2	Stato dell'arte										
	2.1	Sisten	ni robotici riabilitativi e controllo	15							
		2.1.1	Controllo Elettromiografico	15							
		2.1.2	Robot riabilitativi	16							
	2.2 Manipolatori riabilitativi										
		2.2.1	MIT-MANUS	19							
		2.2.2	MIME - Mirror Image Movement Enhancer	20							
	2.3	Sisten	ni Esoscheletrici riabilitativi	21							
		2.3.1	Saga University - Esoscheletri per l'arto superiore	21							
		2.3.2	ETH Zurigo - ARMin	25							
		2.3.3	CADEN-7	26							
		2.3.4	National Taiwan University - Esoscheletro per l'arto								
			superiore	28							
		2.3.5	Waseda University - Ortesi robotica per il gomito	28							
3	Tecniche Elettromiografiche 31										
	3.1 Anatomia del braccio e della spalla										
		3.1.1	La spalla	31							
		3.1.2	Il braccio	34							
	3.2	Origin	e del segnale elettromiografico	36							
		3.2.1	Il muscolo e la sua fisiologia	36							
		3.2.2	Il segnale emg superficiale	41							
	3.3 Strumenti e metodologie di misura										
		3.3.1	Elettrodi e preparazione della cute	45							
		3.3.2	Amplificatore	48							
		3.3.3	Artefatti e disturbi	50							
	3.4	Tratta	amento dei segnali ed estrazione delle features	53							
		3.4.1	Campionamento e filtraggio segnali	53							
		3.4.2	Estrazione features dal segnale emg	53							

4	Stru	ımenti e Metodi	57					
	4.1	L'esoscheletro leggero L-Exos	57					
		4.1.1 Descrizione	57					
		4.1.2 Cinematica diretta	58					
		4.1.3 Aspetti costruttivi e meccanici	60					
	4.2	Segnali elettromiografici	67					
		4.2.1 Amplificatore g.USBamp	69					
		4.2.2 Eliminazione disturbo ECG	70					
	4.3	Schema di acquisizione	76					
5	Pre	dizione di coppia con elettromiografia	79					
	5.1	Perchè usare le reti neurali	79					
	5.2	Funzionamento delle reti neurali	80					
		5.2.1 Modello matematico del Neurone	80					
		5.2.2 Architettura di una rete neurale	82					
		5.2.3 Reti Feedforward	84					
		5.2.4 Addestramento delle reti Feedforward	86					
		5.2.5 Condizioni e regole sul dataset e l'architettura delle reti	98					
	5.3	Dataset Acquisiti	104					
	5.4	Addestramento delle reti neurali per la predizione della coppia	106					
		5.4.1 Risultati sul dataset noGC	112					
		5.4.2 Risultati sul dataset con GC	126					
		5.4.3 Discussione ed analisi dei risultati	140					
6	Controllo in posizione e validazione 1							
	6.1	Controllo in posizione	143					
		6.1.1 Schema di controllo	143					
		6.1.2 Risultati e validazione del controllo in posizione	146					
7	Conclusioni e lavori futuri 15							
Bi	bliog	grafia 1	54					
\mathbf{A}	A Codice Matlab 16							
	A.1	Rimozione del crosstalk ECG	165					
		A.1.1 estimateh.m	165					
		A.1.2 ecgcorr.m	166					
	A.2	Creazione Dataset	166					
		A.2.1 dataset_movimenti.m	166					
	A.3	Addestramento reti TDNN	168					
		A.3.1 meantorque.m	168					

	A.3.2	MAV.m			•					169
	A.3.3	dataset_e_addestramento.m $\ .$.			•					169
	A.3.4	filtro_ecg.m		•	•			•		176
	A.3.5	plot_prestazioni.m			•					176
A.4	Interfa	ccia grafica per l'acquisizione			•					177
	A.4.1	$EMG_Lexos_GUI_isometric_light$		•	•			•		177

Capitolo 1 Introduzione

L'ictus è una delle patologie più comuni che, con l'aumento dell'età media della popolazione, tende a colpire frequentemente la popolazione mondiale. In particolare è stato stimato che solo negli U.S. ogni anno son state colpite circa 795000 persone [1]. Le statistiche italiane mostrano che il 30% dei pazienti colpiti soffrono di disabilità che limitano lo svolgimento delle attività quotidiane [2], tra le quali le più comuni sono le emiparesi dell'arto superiore [3]. L'obiettivo principale della terapia riabilitativa in questi casi è quello di recuperare le funzioni motorie, di migliorare la coordinazione dei movimenti o di insegnare nuove strategie motorie. Uno degli approcci più efficaci per il recupero funzionale degli arti consiste nell'esecuzione di movimenti attivi altamente intensivi e ripetitivi. Le sessioni di riabilitazione tradizionale richiedono un'interazione intensa tra il paziente ed il fisioterapista e peccano di scarsa ripetibilità e di misure poco oggettive del miglioramento delle capacità motorie del paziente. In questo contesto si inseriscono gli studi della comunità scientifica sulla riabiltazione assistita da sistemi robotici. E ormai consolidata l'evidenza sperimentale che la riabiltazione assistita da sistemi robotici possa competere e accelerare i miglioramenti dei pazienti rispetto alle tecniche riabilitative convenzionali [4][5], attraverso sequenze di movimenti ottimizzati e grazie ad incentivi come l'interazione con giochi e realtà virtuale, portando alla riduzione del lavoro intensivo, consentendo nuove modalità di esercizio non attualmente disponibili e portando alla riduzione dei costi di assistenza sanitaria con degenze di durata più breve.

Le prime generazioni di sistemi riabilitativi che permettevano l'interazione con l'utente impostavano l'interfaccia di interazione uomo-macchina (Human-Machine Interface, HMI) al livello dei segnali di posizione o di forza. il controllo dei sistemi robotici in questo caso richiede che l'operatore possa muovere il proprio arto ed applicare degli spostamenti o delle forze sul sistema robotico stesso, cosa non sempre garantita, come nei casi di pazienti affetti da patologie neuro-muscolari, non in grado di completare il movimento. L'idea dominante della ricerca attuale è di impostare l'interfaccia uomo macchina ad un livello gerarchico più alto del sistema fisiologico neurologico, con la possibilità di superare il ritardo il ritardo tra l'intenzione del movimento e l'esecuzione del movimento stesso, dovuto alla trasmissione dei segnali nervosi ed alla dinamica del sistema muscolo-scheletrico. Nel campo del controllo dei sistemi robotici riabilitativi, l'utilizzo dei segnali elettromiografici di superficie (sEMG), permette di poter ottenere dei segnali di controllo intensamente correlati con l'intenzione del movimento, e dunque di superare le limitazioni precedentemente indicate.

L'obiettivo di questo lavoro di tesi è quello di sviluppare un'interfaccia uomo macchina basata su segnali elettromiografici superficiali ed, infine implementare un controllo per l'esoscheletro L-Exos per la riabilitazione dell'arto superiore destro. Lo svolgimento del percorso per il raggiungimento di tali obiettivi è stato svolto presso il laboratorio PERCRO dell'istituto TeCIP della Scuola Superiore Sant'Anna, specializzato nello sviluppo di interfacce aptiche e ambienti di realtà virtuale. In particolare son state portate avanti due attività. Nella prima è stato implementato e validato un sistema di stima delle coppie ai giunti di spalla e gomito basata su segnali sEMG per contrazioni muscolari in condizioni isometriche.

Nella seconda attività, il suddetto sistema di stima delle coppie ai giunti è stato utilizzato per il controllo elettromiografico on-line di esoscheletri ed infine validato.

Più in dettaglio, nel capitolo 2 viene descritto lo stato dell'arte del controllo elettromiografico, ed i sistemi riabilitativi, focalizzando l'attenzione principalmente su quelli controllati mediante i segnali sEMG. Nel capitolo 3 viene descritta l'anatomia dell'arto superiore, la fisiologia della contrazione muscolare e l'origine del segnale elettromiografico. Nel capitolo 4 vengono descritti gli strumenti ed i metodi di misura utilizzati durante il lavoro di tesi, ed in particolare verranno esposti i dettagli tecnici dell'esoscheletro leggero L-Exos sul quale è stato implementato il controllo studiato ed, il sistema di misura dei segnali elettromiografici ed i metodi utilizzati per il loro trattamento. Nel capitolo 5 vengono descritte le reti neurali utilizzate per l'identificazione della coppia articolare a partire dai segnali elettromiografici, le tecniche di addestramento, i dataset raccolti durante la parte di lavoro sperimentale ed i risultati ottenuti sulla stima delle coppie ai giunti. Nel capitolo 6 infine viene proposto un metodo di controllo del sistema esoscheletrico L-Exos a partire dal sistema di stima delle coppie ai giunti descritto nel capitolo precedente.

Il lavoro fatto durante il percorso di tesi è stato inoltre accettato per la conferenza Haptic Symposium 2014 e verrà pubblicato con il titolo "An EMG-based approach for on-line predicted torque control in robotic-assisted rehabilitation".

Capitolo 2

Stato dell'arte

In questo capitolo viene presentato lo stato dell'arte per quanto riguarda i sistemi robotici sviluppati nell'ambito della ricerca della robotica riabilitativa dell'arto superiore, i manipolatori basati sull'end-effector ed i sistemi esoscheletrici. Viene inoltre presentato lo stato dell'arte per quanto riguarda il controllo elettromiografico.

2.1 Sistemi robotici riabilitativi e controllo

Gli avanzamenti della ricerca e della tecnologia nei campi dell'ingegneria meccanica, biomedica e dell'intelligenza artificiale hanno permesso negli ultimi 20 anni di sviluppare sistemi robotici capaci di competere, in campo riabilitativo, con la riabilitazione tradizionale. Molti studi hanno mostrato l'evidenza che le terapie robot-assistite aumentano l'efficacia del trattamento post-traumatico e incentivano il paziente introducendo incentivi al paziente attraverso l'utilizzo della realtà virtuale e di giochi. L'uso di dispositivi robotici nella riabilitazione può offrire trattamenti attivi altamente intensivi, ripetitivi ed interattivi rispetto alla riabilitazione tradizionale. In aggiunta permette di applicare facilmente nuovi vincoli per ottimizzare la sequenza di movimenti richiesta, oltre che a controllarne e monitorarne la precisione, tramite misure oggettive, durante movimenti complessi.

2.1.1 Controllo Elettromiografico

L'interazione tra il sistema robotico ed il paziente può diventare attiva grazie all'interfaccia uomo-macchina (Human-machine Interface, HMI) che permette di utilizzare sistemi hardware-software per il controllo naturale del sistema robotico. In base al sistema robotico ed all'applicazione per il quale è stato sviluppato, son stati proposti numerosi algoritmi per il controllo come ad esempio, il controllo in posizione, il controllo in forza o impedenza. Il moto dei sistemi robotici in questi caso richiede che l'operatore possa muovere il proprio arto ed applicare degli spostamenti o delle forze sul sistema robotico stesso, cosa non sempre garantita, come nei casi di pazienti affetti da patologie neuro-muscolari, non in grado di completare il movimento. Inoltre, la sequenza di comando (di posizione o forza) implica dei ritardi di esecuzione del movimento da parte del robot. L'utilizzo dei segnali elettromiografici (EMG) nel campo del controllo dei sistemi robotici permette di impostare l'HMI ad un livello più alto, ovvero quello neuro-muscolare, permettendo di minimizzare il ritardo elettro-chimico-meccanico tipico dei sistemi muscoloschelettrici. In letteratura, l'analisi dei segnali elettromiografici è stata condotta per realizzare principalmente due tipi di controlli robotici. Il primo permette di classificare i movimenti eseguiti e dunque di utilizzare il segnale EMG come trigger per l'assistenza robotica, il secondo permette un controllo continuo del movimento durante tutta la durata della sua esecuzione. Nel primo caso sono stati condotti numerosi studi sulla classificazione dei movimenti a partire dai segnali EMG tramite differente tecniche, come la Linear Discriminant Analysis [6], Support Vector Machines [7][8], reti neurali [9], tecniche statistiche e fuzzy [10], Gaussian mixture models [11]. Nel secondo caso l'obiettivo principale è quello di legare delle caratteristiche del segnale elettromiografico alla postura dell'arto o alle forze che il sistema muscolo-scheletrico è in grado di esercitare. Per far ciò bisogna suddividere gli approcci utilizzati in due classi, la prima è l'identificazione del legame tra EMG-coppia/forza/posizione mediante un modello del sistema muscolo-scheletrico [12], la seconda ha invece un approccio *model-free* e la relazione è identificata principalmente attraverso l'utilizzo di reti neurali [13][14] [15][16][17] o approcci ibridi basati sulle tecniche neuro-fuzzy proposte da Kiguchi in [18]. Un altro lavoro rappresentativo sul controllo continuo dei movimenti è riportato in [19], dove Artemiadis e Kyriakopoulos identificano un modello dinamico lineare tra EMG e angoli articolari, tenendo in conto sia il concetto di sinergia muscolare [20] che quello delle primitive del movimento [21].

2.1.2 Robot riabilitativi

Per quanto riguarda i sistemi robotici è necessario distinguerli in due categorie, le interfacce aptiche basate sull'end-effector e quelli esoscheletrici. I primi muovono gli arti inducendo delle forze solo sulla mano del paziente, in contrasto con i sistemi esoscheletrici che son caratterizzati dal fatto che gli assi dei giunti sono allineati con quelli anatomici dell'utente. I principali vantaggi nello sviluppo dei sistemi basati sull'end-effector sono la facile adattabilità alle differenti lunghezze degli arti e la semplicità costruttiva del sistema meccanico, che si basano su semplici manipolatori seriali. Lo scopo dei sistemi esoscheletrici deriva invece da motivazioni che prevedono che la struttura robotica sia sempre in contatto con l'arto dell'utente. La condizione



Figura 2.1: Confronto tra lo schema cinematico di un manipolatore classico ed un sistema esoscheletrico.

di contatto fisico tra un sistema esoscheletrico ed il corpo umano non è relativo ad un singolo punto di contatto, come accade per le interfacce aptiche che presentano uno stilo all'end-effector che viene afferrato dall'utente. Il link fisico tra l'esoscheletro e il corpo umano si riferisce ai vari punti di attacco, di solito almeno uno per ogni arto. I vantaggi derivanti da questo approccio sono che la postura dell'arto è sempre completamente determinata e che è possibile evitare estensioni eccessive tramite semplici finecorsa meccanici.

Un esoscheletro pone dei vincoli cinematici agli arti collegati ad esso. In particolare il braccio umano ha 7 gradi di libertà: abduzione/adduzione e flessione/etensione della spalla, rotazione del braccio, flessione/estensione del gomito, rotazione dell'avambraccio, deviazione ulnare/radiale e la flessione/estensione del polso. É auspicabile che il dispositivo esoscheletrico non comprometta il movimento naturale dell'arto e l'workspace dello stesso. Inoltre deve anche avere capacità di coppia tali da soddisfare o migliorare le capacità umane. La tabella 2.1 mostra l'workspace ed i limiti di coppia dell'arto superiore umano.

Il design meccanico dei sistemi esoscheletrici comprende molte limitazioni dovute allo stato dell'arte degli attuatori, alla complessità dei giunti articolari umani ed ai materiali usati per la loro costruzione. Molte interfacce

Giunto	Forza isometrica (Nm)	Workspace giunto (gradi)						
Spalla (Abduzione/Adduzione)		Abduzione: 150 Adduzione: 0						
Spalla (Flessione/Estensione)		Flessione: 180 Estensione: 60						
Spalla (Rotazione interna/esterna)		Rotazione interna: 90 Rotazione esterna: 90						
Gomito (Flessione/Estensione)	72.5	Flessione: 146 Estensione: 0						
Avambraccio (Supinazione/Pronazione)	9.1	Supinazione: 86 Pronazione: 71						
Polso (Flessione Dorso-Palmare)	19.8	Flessione palmare: 73 Flessione dorsale: 71						
Polso (Abduzione/Adduzione)	20.8	Adduzione: 33 Abduzione: 19						

Tabella 2.1: Workspace e limiti di coppia dell'arto superiore umano.

esoscheletriche tentano di ottimizzare una o più delle seguenti caratteristiche dei sistemi aptici:

- rapporto potenza/peso
- workspace
- indossabilità
- stabilità
- ampiezza di banda del controllo

Il design individuale permette di ottenere tali ottimizzazioni, tuttavia molto spesso a discapito di altre importanti, come ad esempio dell'workspace. Basti pensare che la grandezza e la forma del workspace stesso è tipicamente dipendente dall'applicazione obiettivo ed è un fattore importante per determinare la dimensione complessiva del manipolatore e dei meccanismi. Workspace più ampi vengono ottenuti a discapito di maggiore complicatezza della struttura del manipolatore e quindi del relativo peso e dimensioni e dunque della mobilità dell'esoscheletro.

Di seguito sono elencati brevemente alcuni esempi dell'attuale stato dell'arte nel campo dei sistemi robotici riabilitativi. In particolare, le difficoltà di controllo real time di sistemi robotici mediante elettromiografia, fanno si che sul mercato esistano pochi sistemi assistivi o riabilitativi controllati attraverso i segnali EMG.

2.2 Manipolatori riabilitativi

2.2.1 MIT-MANUS

Il MIT-Manus [22] è un manipolatore robotico riabilitativo sviluppato nel 1991 al Massachusetts Institute of Technology (MIT) e poi prodotto nella sua versione commerciale InMotion2. I due sistemi riabilitativi hanno permesso la riabilitazione dell'arto superiore di oltre 200 pazienti colpiti da ictus.

La struttura cinematica del robot è planare di tipo SCARA che permette al paziente di muovere la mano lungo il piano trasversale anteriore del corpo con 2 DOF. La prima versione del sistema robotico prevede un endeffector con uno stilo per la presa della mano, mentre nelle versioni successive è stato previsto un ulteriore meccanismo [23] per il movimento e la riabilitazione dell'articolazione del polso, permettendo di avere 5 DOF attivi. La struttura meccanica del manipolatore è stata concepita per avere bassa impedenza all'end-effector, e dunque di essere percepito minimamente dall'utente. L'utilizzo del MIT-manus prevede delle applicazioni video-giochi per incenti-



Figura 2.2: MIT Manus nella sua versione commerciale InMotion2, manipolatore per la riabilitazione di spalla e gomito.

vare l'utente durante la sessione di riabilitazione. Il controllo implementato nel sistema robotico è un controllore di impedenza all'end-effector, ma è stato sperimentato un controllo basato sui segnali elettromiografici per il trigger del movimento. In particolare in [24] è descritto l'algoritmo implementato per il controllo. Lo spazio di lavoro per la riabilitazione è diviso in 8 settori angolari, equispaziati di 45° secondo i punti cardinali. L'obiettivo del task motorio è l'estenzione del braccio lungo queste direzioni. Il manipolatore esegue il movimento secondo l'intenzione di movimento del paziente, identificata in base al superamento di una soglia di attivazione muscolare per i muscoli principali coinvolti nel movimento stesso.

2.2.2 MIME - Mirror Image Movement Enhancer

Il MIME (Mirror Image Movement Enhancer) [25][26] è un sistema robotico per la riabilitazione del gomito e della spalla per movimenti complessi 3D, sviluppato dal Department of Veterans Affairs di Palo Alto U.S. e dal Division of Physical Medicine and Rehabilitation, Department of Functional Restoration, Stanford University. L'idea di fondo del MIME è la possibilità di riabilitare l'arto emiparetico attraverso il movimento dell'arto sano. Il sistema riabilitativo è composto complessivamente da due parti, un manipolatore robotico industriale Puma-560 a 6 DOF, il cui end-effector ha un supporto per la mano dell'utente, con maniglia per la presa della mano, e da un altro manipolatore utilizzato per la misura dei movimenti della mano dell'arto sano. L'end-effector del Puma-560 è dotato di un sensore di forza a 6 assi per la misura delle forze e delle coppie dovute all'interazione tra l'arto emiparetico ed il robot stesso. Il sistema può essere utilizzato in 4 moda-



Figura 2.3: MIME (Mirror Image Movement Enhancer), sistema robotico per la riabilitazione di spalla e gomito .

lità differenti, passivo, attivo-assistito, attivo-vincolante e bilaterale. Nella modalità passiva l'arto emiparetico compie delle traiettorie predeterminate,

in quella attiva-assistiva l'utente interagisce con il manipolatore con movimenti volontari che vengono assistiti dal sistema riabilitativo, nella modalità attiva-vincolante il robot fornisce una resistenza viscosa nella direzione del movimento desiderato, in quella bilaterale il soggetto esegue dei movimenti bilaterali attraverso l'utilizzo del braccio sano ed il sistema riabilitativo mantiene la simmetria a specchio tra i due avambracci.

Il sistema non è disponibile in versione commerciale, ma è stato utilizzato per la riabilitazione di circa 30 pazienti colpiti da ictus.

2.3 Sistemi Esoscheletrici riabilitativi

2.3.1 Saga University - Esoscheletri per l'arto superiore

Nel 1997 nella Saga University in Giappone inizia il programma di ricerca nel campo degli esoscheletri controllati con l'elettromiografia, portando a sviluppare col tempo una serie di prototipi ad iniziare dall'esoscheletro ad 1 DOF [27] per la riabilitazione del gomito, permettendo flessione ed estensione dell'avambraccio con un controllo di tipo Neuro-fuzzy tramite segnali elettromiografici del braccio e un sensore di forza al polso. Successivamente lo stesso concetto è esteso all'articolazione della spalla, portando a concepire un esoscheletro per la sua riabilitazione [28] permettendo il movimento dei tre gradi di libertà della spalla, l'estensione e la flessione sul piano saggittale, l'estensione e l'estensione orizzontale sul piano traverso e la abduzione-adduzione della spalla. La fusione dei due esoscheletri ha portato alla realizzazione di un unico robot riabilitativo a 4 DOF [29] ed infine alla realizzazione del seguente Sueful-7 [30] raffigurato in 2.6, esoscheletro a 7DOF studiato per essere indossato dal braccio destro come mostrato dalla 2.6, capace di assistere il movimenti orizontali e verticali di flessione/estensione della spalla, le sue rotazioni interne/esterne, la flessione/estensione del gomito, la supinazione/pronazione dell'avambraccio, la flessione/estensione del polso e le deviazioni ulna/radiali del polso. La struttura dell'esoscheletro ha un basso peso (circa 5 Kg) che consente di poterlo installare su una sedia a rotelle e soprattutto di minimizzare la percezione dello stesso sul braccio dell'utente. Ogni giunto è attuato da motori elettrici e la trasmissione del movimento al link è permessa da ingranaggi cilindrici. Ogni giunto è sensorizzato con potenziometri per la misura della posizione, sul gomito e sul polso son presenti dei sensori di forza. La distanza tra il sostegno del braccio ed il centro di rotazione del giunto della spalla dell'esoscheletro è regolabile in base alle esigenze anatomiche del paziente in modo da cancellare il più possibile gli effetti negativi dovuti la



Figura 2.4: Esoscheletro 1DOF per il gomito

distanza tra la spalla del robot e quella dell'utente. Il workspace del robot viene limitato da fermi meccanici e fine corsa per evitare di oltrepassare i limiti anatomici del paziente, mentre le velocità e le coppie ai giunti vengono limitate dal software per prevenire movimenti inaspettati o troppo rapidi.

L'esoscheletro per il gomito descritto in [27] viene controllato secondo le intenzioni del movimento eseguito dal gomito dell'utente, secondo delle relazioni che descrivono il comportamento muscolare-articolare, modellato con relazioni di tipo EMG-impedenza ai qiunti e movimenti angolari desiderati o Forza generata al polso-impedenza ai giunti e movimenti angolari desiderati, entrambe ottenute tramite regole fuzzy IF-THEN. Vengono utilizzate due relazioni in quanto per bassi livelli di attivazione muscolare, l'informazione di forza presente all'interno dei segnali elettromiografici è difficilmente utilizzabile, perchè poco significativa, e dunque viene reso necessario l'utilizzo dei sensori di forza. Per medie attivazioni viene utilizzata la tecnica della sensor fusion, con la quale entrambe le informazioni elettromiografiche e di forza vengono utilizzate per il controllo dell'esoscheletro, mentre per alte attivazioni vengono utilizzati solo i segnali elettromiografici. I segnali elettromiografici utilizzati vengono misurati sul lato mediale e laterale di bicipite e tricipite, e dai 4 segnali vengono estratte delle features. In questo studio la feature usata per ciascun segnale è la Waveform Lenght, ovvero la lunghezza cumulativa della forma d'onda in un segmento temporale di 50ms, utilizzata per esprimere l'attivazione secondo tre livelli fuzzy (zero, positivo piccolo, positivo grande). Il controllo implementato è di tipo neuro-fuzzy, in modo tale da adattare (mediante backpropagation online) il controllo alle condi-



Figura 2.5: Esoscheletro 3DOF per la spalla



Figura 2.6: SUEFUL-7: Esoscheletro a 7DOF per l'assistenza del moto dell'arto superiore

zioni fisiologiche del soggetto che utilizzata l'esoscheletro. Le uscite generate dalla rete neuro-fuzzy sono gli angoli di giunto desiderati e l'impedenza del robot esoscheletrico.

L'esoscheletro descritto in [28] ha un controllo simile a quello del gomito, una struttura neuro-fuzzy con in ingresso i segnali elettromiografici dei gruppi muscolari interessati dalla spalla e i due angoli, verticale ed orizzontale della spalla stessa. I segnali elettromiografici son stati acquisiti dai muscoli deltoide (anteriore, posteriore e laterale), dal pettorale maggiore (mediale e parte clavicolare), il grande rotondo ed il trapezio. Da questi 7 segnali emg vengono estratte le features, il valore assoluto medio (MAV) per una finestra di lunghezza 50ms, ed infine vengono utilizzate per definire dei livelli fuzzy di attivazione (zero, positivo piccolo, positivo medio, positivo grande) utilizzati assieme agli angoli di spalla in ingresso alla rete neuro-fuzzy. I valori della rete vengono adattati per modellare il controllo sulle esigenze fisiologiche del soggetto che utilizza l'esoscheletro, in particolare vengono modificati i valori dei segnali elettromiografici tramite dei coefficienti correttivi che tengono in considerazione il diverso utilizzo dei gruppi muscolari in base agli angoli di spalla. L'uscita della rete neuro-fuzzy è la coppia di controllo dei giunti dell'esoscheletro, utilizzata per generare gli angoli di spalla desiderati tramite il controllo ad impendenza.

La terza versione dell'esoscheletro riabilitativo [29] include entrambi i controlli sulla spalla e sul gomito degli esoscheletri precedenti, con una modifica sulla struttura dello stesso, sempre di natura neuro-fuzzy ma stratificato in tre blocchi funzionali: un primo stadio per la selezione dei segnali, un secondo per la selezione della postura ed un terzo con il controllo vero e proprio. Nel primo viene applicato un controllo basato sull'attivazione muscolare o sul sensore di forza al polso, in base al livello di attività muscolare dell'utilizzatore del robot. Nel secondo viene scelto il controllore neuro-fuzzy opportuno in base agli angoli di spalla e gomito, suddivisi in tre regioni per tenere in considerazione la nonlinearità del sistema muscolare. Nel terzo stadio per ciascun giunto viene calcolata la coppia necessaria per realizzare la corretta assistenza che il robot dovrà dare all'utente.

Nella versione definitiva dell'esoscheletro SUEFUL-7 [30] il controllo implementato è il medesimo dei casi precedenti, con i segnali elettromiografici acquisiti per 16 muscoli, dai quali vengono estratte delle features utili per il controllo. In questo studio le features utilizzate sono i valori RMS del segnale elettromiografico per una finestra di segnale dello stesso. La coppia stimata ai giunti, attraverso il segnale elettromiografico viene stimata attraverso una combinazione lineare di tutti i segnali elettromiografici dei muscoli coinvolti nell'articolazione, con pesi che vengono corretti tramite le opportune leggi neuro-fuzzy in base alla postura dell'arto, ed una calibrazione online per adattare il controllo alle condizioni fisiologiche dell'utente. Gli stessi parametri d'impedenza utilizzati per il controllo in forza dell'esoscheletro vengono calcolati in base ad opportune leggi neuro-fuzzy dipendenti dall'attivazione muscolare dei muscoli agonisti del movimento e dalla postura dell'arto.

2.3.2 ETH Zurigo - ARMin

L'ARMin [31] è il semi-esoscheletro sviluppato all'ETH di Zurigo in collaborazione con l'University Hospital Balgrist, la cui versione attuale ha 4 gradi di libertà attuati ed altri due passivi, in modo da permettere flessione/estensione del gomito ed i movimenti spaziali della spalla con altri tre gradi di libertà sui movimenti di flessione/estensione e di adduzione e abduzione. Il robot è fissato su una struttura di alluminio e la postazione del paziente è bloccata in posizione seduta su una sedia a rotelle appena sotto il robot stesso, concepito per far eseguire al paziente dei movimenti ripetitivi con feedback visibi, auditivi e aptici. La struttura cinematica con i 4DOF attuati permette di posizionare la mano del paziente in tutto il workspace del braccio umano, permettendogli di lavorare non solo sul piano trasversale ma anche su quello saggittale, consentendo una riabilitazione più funzionale per afferrare e manipolare oggetti. Il robot può essere esteso per permettere pronazione/supinazione dell'avambraccio e flessione/estensione dei movimenti del polso, mentre la deviazione ulnare/radiale è vincolata ad una posizione neutra a causa della minore importanza funzionale di questi movimenti e per ridurre la complessità del dispositivo. La presenza di gradi di libertà non attuati permette la riabilitazione in sicurezza del paziente, preservandone il naturale range dei movimenti, limitati alla mano con velocità di 1 m/s ed accelerazioni di 10 m/s^2 . La strumentazione del robot è dotata di encoder angolari e lineari per la misura delle posizioni dei giunti e di sensori di forza/coppia permettendo il controllo ad impedenza variabile in base al task e la compensazione della dinamica del robot stesso.

Il robot consente tre tipi di terapie al paziente. La *Movement Therapy* il cui obbiettivo è di prevenire la degenerazione delle articolazioni umane e di preservarne la mobilità. In questa modalità il terapista può guidare il braccio umano mentre il robot registra le posizioni. Una volta registrata la traiettoria può essere ripetuta ed eseguita dal robot con diverse velocità. La modalità *Game Therapy* serve a motivare il paziente con semplici giochi rappresentati in audio-video su uno schermo. In questa modalità il robot supporta il paziente nell'esecuzione del movimento con quanta forza è necessaria. Nella modalità *ADL Training* il paziente può esercitarsi nelle attività quotidiane come il mangiare o afferrare oggetti. Il robot misura la posizione e la velocità per predirre le forze e le coppie utilizzando un modello meccanico del paziente e del robot stesso, ed infine assiste il paziente nel movimento in base ad un fattore di sostegno per fornire le forze e le coppie necessarie a completare il movimento.

Una nuova versione dell'Armin2 [32] è stata realizzata ed è attualmente in fase di test sui pazienti.



Figura 2.7: Armin: Struttura meccanica, Figura 2.8: Armin: Versiosensori e motori. ne attuale dell'esoscheletro e postazione paziente.

2.3.3 CADEN-7

I lavori di Rosen nel campo della robotica riabilitativa hanno portato a sviluppare alcune tecniche di controllo basate sui segnali elettromiografici, utili per il controllo di sistemi esoscheletrici dell'arto superiore. In particolare nei primi lavori l'attenzione è stata focalizzata sullo sviluppo di un sistema di controllo capace di predirre la coppia articolare del gomito a partire dai segnali elettromiografici dei principali gruppi muscolari del braccio. L'approccio utilizzato è basato sul modello di Hill [12] per il calcolo delle coppie articolari a partire dai segnali elettromiografici. Il sistema di controllo così sviluppato è stato utilizzato nell'esoscheletro per l'arto superiore descritto in [33], esoscheletro a due gradi di libertà, per l'estensione/flessione della spalla, e l'estensione/flessione del gomito. Il sistema esoscheletrico sviluppato è essenzialmente un dispositivo di assistenza, capace di amplificare il momento generato dalle contrazioni muscolari durante la manipolazione di carichi. I giunti sono attuati mediante servomotori DC e riduttori planetari per ridurne la coppia e sensorizzati con encoder incrementali. Il sistema è stato concepito per essere trasportabile e di basso peso, ed essere utilizzabile per l'assistenza medica di utenti disabili.

Il sistema di controllo sul DOF del gomito descritto in [33], è stato sviluppato nei lavori successivi [34] per il controllo dei 3 DOF dell'articolazione del polso. É prevista l'integrazione del sistema di controllo nel CADEN-7



Figura 2.9: CADEN-7, Esoscheletro 7 DOF.

[35], esoscheletro a 7 DOF per il braccio destro. La progettazione meccanica del CADEN-7 è stata portata avanti per perseguire due obiettivi principali, la massimizzazione del workspace dell'arto umano e la sicurezza del suo utilizzo. In particolare i giunti dell'esoscheletro son tutti di natura antropomorfica, caratteristica che ha permesso di aumentare il workspace in maniera sensibile rispetto agli altri esoscheletri allo stato dell'arte, senza aumentare la complessità del sistema meccanico con aggiunte di ulteriori gradi di libertà. Il risultato è un esoscheletro con parti mobili di peso pari a 6.8 kg e con workspace pari al 99% di quello richiesto per completare attività quotidiane (ADL). La sicurezza durante l'utilizzo dell'esoscheltro è garantita da fermi meccanici che non permettono di superare le ampiezze massime possibili dei giunti articolari umani, da pulsanti di arresto di sicurezza e limiti software sulle coppie. Il moto dei link dell'esoscheletro è ottenuto attraverso motori DC posizionati sulla base dell'esoscheletro (link fisso) e la trasmissione del moto ai giunti è ottenuta attraverso un sistema di trasmissione a tendini e pulegge.

I sensori di posizione sono ridondanti, ogni giunto è sensorizzato con potenziometri e i motori vengono sensorizzati con encoder, in modo tale da monitorare continuamente l'integrità del sistema di trasmissione.

2.3.4 National Taiwan University - Esoscheletro per l'arto superiore

L'esoscheletro progettato e sviluppato alla National Taiwan University [36] è un manipolatore robotico con 9 DOF, che includono 6 DOF al complesso di giunti della spalla, 1 DOF al gomito e 2 DOF al giunto del polso. I gradi di libertà del sistema esoscheletrico son maggiori rispetto a quelli dell'arto umano (7 DOF), rendendo la struttura ridondante, per la necessità di ampliare il più possibile il workspace. La Struttura dell'esoscheletro è adattabile alla lunghezza dell'arto dell'utente. Ogni giunto è sensorizzato tramite potenziometri o gli encoder dei motori. I punti di contatto con l'arto umano sono inoltre sensorizzati attraverso 4 sensori di forza per misurare le interazioni tra l'utilizzatore ed il robot. I giunti sono attuati mediante motori DC Faulhaber, con alto rapporto di coppia/peso. Il sistema di controllo prevede due modalità di riabilitazione, una attiva ed una passiva, ed integra all'interno un controllo PID, un controllo di impedenza ed un trigger EMG. Il controllo PID ha il compito di annullare l'errore di posizione del sistema esoscheletrico, il controllo di impedenza ha il compito di minimizzare il più possibile l'interazione di forza tra il sistema esoscheletrico e l'utente. Il sistema di controllo elettromiografico implementa una modifica del sistema del MIT-Manus [24]. Attraverso la comparazione tra il segnale EMG e quello di forza, il sistema è in grado di rilevare il giusto grado di contrazione muscolare e di fornire all'utente l'ausilio per completare il movimento. In particolare viene calcolata l'attività mioelettrica per ciascun muscolo, come radice della media quadratica (RMS) e confrontata con il valore di soglia del muscolo a riposo. Per ciascun movimento viene identificato uno specifico muscolo, responsabile del movimento stesso. Se l'attività mioelettrica dello specifico muscolo supera la soglia a riposo, il sistema esoscheletrico fornisce l'ausilio con la coppia necessaria.

2.3.5 Waseda University - Ortesi robotica per il gomito

L'ortesi robotica descritta in [37] è un prototipo di supporto per i movimenti del gomito, capace di sopprimere il tremore dell'arto dovuto a patologie o all'anzianità dell'utilizzatore. La struttura meccanica è composta da due link ed un giunto attuato attraverso un motore DC con Harmonic drive system con rapporto di riduzione 1:100, utile per la soppressione del tremore. La struttura dell'ortesi ha un peso complessivo 330 g, i link sostengono il gomito attraverso delle fasce con velcro. La soppressione del tremore del paziente avviene attraverso il riconoscimento delle fasi di contrazione volontaria attraverso i segnali elettromiografici e time delay neural networks (TDNN), adde-



Figura 2.10: Esoscheletro 9DOF costruito alla National Taiwan University a) struttura meccanica b) catena cinematica del manipolatore esoscheletrico.

strate per il riconoscimento degli stati ON OFF della contrazione volontaria rispetto alle contrazioni dovute al tremore.



Figura 2.11: Ortesi anti-tremore sviluppata alla Waseda University.

Capitolo 3

Tecniche Elettromiografiche

In questo capitolo viene descritta brevemente l'anatomia del braccio e della spalla umana, con i relativi muscoli interessati nell'esecuzione dei movimenti descritti in questo lavoro di tesi. Viene inoltre descritta l'origine del segnale elettromiografico, i metodi e le apparecchiature di misura utilizzate, le caratteristiche del segnale ed i disturbi che deteriorano la qualità della misura. Nell'ultima sezione vengono descritte le principali features del segnale elettromiografico utili per il controllo di manipolatori e protesi.

3.1 Anatomia del braccio e della spalla

3.1.1 La spalla

La spalla è una delle quattro parti dell'arto superiore che in senso prossimo distale sono: spalla, braccio, avambraccio, mano. La spalla è il dispositivo di attacco dell'arto superiore al tronco, il suo scheletro forma nell'insieme la cintura toracica che si unisce alla parte superiore del torace e da attacco alle restanti parti dell'arto che, nell'insieme formano la cosiddetta parte libera. Se si osserva la conformazione dello scheletro del corpo umano ci si accorge di una cosa piuttosto singolare, ovvero che le ossa che compongono la spalla (omero, clavicola, scapola) son collegate al busto, in particolare allo sterno, solo attraverso l'articolazione con la clavicola. La spalla è tenuta nella sua posizione di equilibrio dai numerosi muscoli che agiscono come coppie motorie sulla spalla, lungo assi di movimento che si possono considerare infiniti ma che sostanzialmente sono riconducibili a tre principali:

• asse sagittale, dove avvengono i movimenti di abduzione ed adduzione dell'arto superiore;

- asse verticale, attorno al quale avviene la rotazione interna ed esterna dell'arto
- asse frontale, lungo cui si esegue l'anteposizione e la retroposizione.

Le articolazioni che partecipano ai movimenti della spalla sono l'articolazione scapolo omerale, l'articolazione acromion clavicolare e l'articolazione sterno clavicolare.

Ciascun cingolo scapolare si compone di tre diverse ossa: una scapola, una clavicola e la parte prossimale dell'omero. Si chiama scapola l'osso triangolare, pari e simmetrico, piatto e sottile con la base superiore e l'apice inferiore applicato alla parte postero superiore del torace (3ř 7ř costa.). Vi si descrivono una faccia anteriore o costale, una faccia posteriore o dorsale, tre margini di cui uno mediale, uno laterale e uno superiore e tre angoli distinti come sopra. La sua funzione consiste, oltre che nel fornire protezione e resistenza al dorso, nel permettere con la sua grande mobilità, i suddetti movimenti di adduzione ed abduzione. Il suo ruolo e pure fondamentale nel movimento di rotazione del braccio attorno al centro costituito dalla spalla, poiché i muscoli che lo causano sono collegati anche ad essa.

La clavicola è un osso lungo, pari e simmetrico a forma di esse disposta trasversalmente, al davanti della prima costa, tra sterno e scapola. Forma la parete anteriore del cingolo scapolare, si collega con la sua parte esterna alla scapola e all'omero e con quella interna allo sterno, l'osso impari centrale della gabbia toracica. Il suo ruolo all'interno dell'organismo è insieme di sostegno dei muscoli e di articolazione della spalla.

L'omero è l'osso che si trova all'interno del braccio, mentre la sua parte più lontana dal tronco (o epifisi distale) si articola con l'ulna e con il radio, formando con il gomito la sua testa, ovvero l'epifisi prossimale, più vicina al tronco, si articola con la clavicola e la scapola all'altezza della spalla. L'omero è essenziale per il moto del cingolo scapolare, in quanto delle tre ossa costituisce quella che visibilmente compie i tre movimenti suddetti.

I movimenti della spalla sono possibili grazie ad un sinergismo tra i muscoli tonici che stabilizzano e mantengono centrata la testa omerale nella glenoide, ed i muscoli fasici che possono conseguentemente muovere la spalla sul fulco così stabilito senza innalzare l'omero. Solo se gli uni stabilizzano la testa omerale, gli altri possono iniziare il corretto complesso movimento della spalla nel suo insieme. Sono muscoli principalmente di stabilizzazione, ma non solo tali, il sopraspinoso, il sottoscapolare, ed il gran pettorale. Sono muscoli principalmente di movimento il deltoide e gli extrarotatori. Ci sono 19 muscoli (considerando 2 romboidei e lo sternocleidmastoideo) che controllano il cingolo scapolare, su un totale di 54 muscoli per tutto l'arto superiore. I muscoli che s'inseriscono sul cingolo scapolare e che quindi controllano i vari movimenti della spalla possono essere classificati in questo modo:

- muscoli della spalla;
- muscoli toraco-appendicolari;
- muscoli spino appendicolari;
- muscoli del braccio.

I principali muscoli presenti nei cingoli scapolare-omerale sono:

- Deltoide è il muscolo pari che ricopre l'articolazione del cingolo scapolare, si unisce sia alla scapola che alla clavicola che all'omero. É in grado di far compiere al braccio diversi movimenti, solleva il braccio di circa 90ř, il fascio anteriore ha un'azione di flessione ed intrarotazione, il medio di massima abduzione sull'omero ed il posteriore un'azione di estensione, abduzione ed extrarotazione. Complessivamente esso è in grado di far compiere al braccio un'abduzione contraendosi e un'adduzione distendendosi. Il movimento di rotazione deriva dal suo moto coordinato con infraspinato e grande pettorale.
- **Trapezio** è un muscolo pari e simmetrico molto vasto, esso orrigina dalle vertebre cervicali e toraciche ma anche sulle clavicole, passando al di sopra della spalla. A causa di quest'ultima origine è in grado contraendosi di far alzare le clavicole e con esse l'intera spalla.
- Infraspinato è il muscolo concorrente del pettorale, si trova nella schiena, parzialmente nascosto alla vista dal trapezio e si collega tramite i tendini sia con la scapola che con l'omero. La contrazione dell'infraspinato comporta la distensione del pettorale, ed il suo ruolo è quello di far muovere il braccio in dietro.
- Sopraspinato è un muscolo pari di forma piramidale, che si origina dalla fossa sopraspinata della scapola per inserirsi nel tubercolo maggiore dell'omero. Il muscolo abduce e ruota all'esterno il braccio in sinergismo con il muscolo deltoide, ed oltre a questo è il principale stabilizzatore della scapolo-omerale.
- Piccolo rotondo è un muscolo pari di forma cilindrica che origina dalla fossa infraspinata della scapola per inserirsi nel tubercolo maggiore dell'omero. La sua contrazione permette di ruotare all'esterno il braccio e di stabilizzare la scapolo-omerale.

- Grande rotondo è un muscolo pari riconoscibile inferiormente al piccolo rotondo; origina dall'angolo inferiore della scapola per inserirsi nel tubercolo minoreS e nel solco intertubercolare. Permette movimenti di adduzione, estensione e rotazione interna dell'omero.
- sottoscapolare è un muscolo pari di forma triangolare, origina dalla faccia anteriore della scapola per inserirsi nel tubercolo minore dell'omero, passando sotto il processo coracoideo. Permette l'adduzione e rotazione interna dell'omero oltre alla stabilizzazione della scapoloomerale.
- Grande pettorale fa parte dei muscoli estrinseci del torace, ha forma di ventaglio ed è composto da tre capi. Origina dal margine anteriore della clavicola, dalla faccia anteriore dello sterno, dalle prime sei cartilagini costali e dalla guaina dei muscoli anteriori dell'addome. I tre capi si inseriscono con un unico tendine alla cresta del tubercolo maggiore dell'omero. Il muscolo grande pettorale permette molte funzioni del braccio: flette (o adduce) in orizzontale, flette (eleva), estende (abbassa), adduce, antepone e ruota internamente l'omero. Prendendo invece punto fisso sull'omero solleva il tronco. I diversi fasci assumono ruoli prevalenti a seconda del piano di lavoro del braccio:
 - Capo Clavicolare: principalmente coinvolto nella flessione (o elevazione) del braccio o nella flessione (adduzione) orizzontale obliqua tendente verso l'alto.
 - Capo sternocostale: il fascio più esteso, interviene in tutti i movimenti ed in particolare sulla pura flessione (adduzione) orizzontale del braccio.
 - Capo Addominale: è maggiormente coinvolto nell'estensione (o abbassamento) del braccio, o nella flessione (o adduzione) orizzontale obliqua tendente verso il basso e adduzione del braccio.

3.1.2 Il braccio

Il braccio è la porzione dell'arto superiore compresa fra le articolazioni della spalla e del corrispondente gomito. Il suo sistema scheletrico è composto unicamente dall'omero, che è già stato brevemente descritto nella sezione precedente. Più in particolare è costituito da un corpo, o diafisi, e da due estremità dette epifisi (prossimale e distale). L'epifisi prossimale si articola con la scapola costituendo l'articolazione scapolo-omerale, mentre l'epifisi distale si articola con le due ossa dell'avambraccio, radio e ulna. La seconda
articolazione nella quale si inserisce, quella del gomito, è un complesso articolare costituito da tre diverse giunzioni: l'articolazione omero-ulnare, l'articolazione omero-radiale e l'articolazione radio-ulnare prossimale. Tutti i capi articolari di queste tre giunzioni sono compresi entro una capsula articolare comune. Le superfici articolari si trovano sull'estremità distale dell'omero e sulle estremità prossimali di radio e ulna. L'articolazione del gomito consente movimenti dell'avambraccio sul braccio, a livello delle due articolazioni omero-ulnare e omero-radiale e movimenti del radio sull'ulna a livello dell'articolazione radio-ulnare prossimale. I principali movimenti dell'avambraccio sono quelli di flessione e di estensione. Essi si svolgono principalmente ad opera dell'ulna che trasporta passivamente il radio. A livello dell'articolazione del gomito si svolgono anche limitati movimenti di lateralità che sono possibili solo quando l'avambraccio è flesso. Lo spostamento del radio rispetto all'ulna con modificazioni nei rapporti degli assi longitudinali è denominato movimento di prono-supinazione.

Per quanto riguarda il sistema muscolare, i muscoli del braccio si distinguono in anteriori e posteriori ed i principali son di seguito elencati:

• Bicipite Brachiale, è un muscolo anteriore che deve il suo nome ai due capi che lo originano, uno lungo ed uno breve, che inferiormente confluiscono in un ventre unico. Il capo lungo origina dalla tuberosità sovraglenoidea della scapola e dal labbro glenoideo mediante un lungo tendine che gioca un ruolo importante nella fisiologia della spalla. Il tendine ha una funzione fondamentale di puleggia stabilizzatrice della testa omerale, in sinergismo con il tendine del sovraspinato. Il capo breve, mediale rispetto al lungo origina dal processo coracoideo della scapola ed unitosi al lungo entrambi s'inseriscono con un robusto tendine sulla tuberosità del radio.

Il bicipite è un muscolo biarticolare, quindi agisce sia sull'avambraccio che sul braccio. Contraendosi flette l'avambraccio sul braccio (principale flessore) e sviluppa ad avambraccio prono una notevole azione supinatoria. Inoltre svolge un'azione stabilizzante a livello dell'articolazione scapolo-omerale, grazie soprattutto al capo lungo, ed è un debole flessore dell'articolazione della spalla. Nell'estensione del gomito il bicipite entra in tensione per contrastare la forza di gravità.

• Coracobrachiale, muscolo anteriore che si trova profondamente al capo breve del tricipite ed origina dal processo coracoideo della scapola. I suoi fasci portandosi in basso s'inseriscono sul terzo medio della faccia anteromediale dell'omero. In particolare si trova posteriormente al bicipite (nella parte prossimale è mediale ad esso) e al grande pettorale. Il muscolo coracobrachiale causa la flessione, la flessione orizzontale, e l'adduzione del braccio.

- Brachiale, è un muscolo anteroinferiore del braccio, posto in profondità rispetto al muscolo bicipite brachiale. Forma la parte superiore del pavimento della fossa cubitale nel gomito. Origina immediatamente al di sotto dell'inserzione del muscolo deltoide, dalle facce antero-mediale e antero-laterale dell'omero, fino all'epifisi distale dell'omero, e si porta in basso per inserirsi sulla tuberosità dell'ulna sulla faccia inferiore del processo coronoideo. È il più potente flessore dell'avambraccio. A differenza dell'altro flessore del"avambraccio, il muscolo bicipite brachiale, non si inserisce sul radio, quindi non partecipa alla supinazione dell'avambraccio.
- Tricipite Brachiale, è il principale muscolo del compartimento posteriore del braccio. Il suo nome deriva dalla sua caratteristica di possedere tre origini tendinee e dunque costituito da tre capi: capo lungo, capo laterale e capo mediale. Il capo lungo origina dalla tuberosità sottoglenoidea della scapola e dal labbro glenoideo, il capo laterale origina dalla faccia posteriore dell'omero al di sopra del solco del nervo radiale, mentre il capo mediale origina dalla faccia posteriore dell'omero al di sotto del solco del nervo radiale. I tre capi convergono in basso in un unico ventre muscolare per andare ad inserirsi sulla faccia posteriore dell'oleocrano dell'ulna.

Il tricipite brachiale è il principale estensore dell'articolazione del gomito in cui è coinvolto prevalentemente il capo mediale, mentre gli altri due fungono da capi accessori, mentre il capo lungo è un adduttore accessorio del braccio.

3.2 Origine del segnale elettromiografico

3.2.1 Il muscolo e la sua fisiologia

Il sistema muscolare è l'insieme di tessuti che permette, attraverso la contrazione muscolare, il movimento del soggetto e lo scorrimento di sostanze organiche interne come sangue e cibo. Nei vertebrati il sistema muscolare è costituito da due tipologie principali di muscoli, i muscoli volontari striati ed i muscoli involontari o lisci, a parte il cuore che pur essendo involontario ha principalmente fibre muscolari striate.

I muscoli volontari, rispetto alla loro collocazione nel corpo son distinti in cutanei e scheletrici. I primi sono localizzati immediatamente sotto la pelle e la loro azione agisce sulla pelle stessa, creandone pieghe e solchi. I muscoli scheletrici hanno le loro principali inserzioni sullo scheletro. Si fissano su organi differenti e mobili l'uno rispetto all'altro, in modo che quando si contraggono, avvicinando l'uno all'altro i punti di attacco, determinano in un cambiamento reciproco di posizione e di conseguenza il movimento. La funzione del muscolo è dunque quella di creare una forza per stabilizzare o per muovere un'articolazione. La forza che il muscolo produce attraverso la contrazione dipende da diversi fattori, ma è comunque direttamente proporzionale alla sezione trasversa del muscolo stesso. Ciascun muscolo è formato da cellule cilindriche allungate, le fibre muscolari. Più fibre muscolari, circondate da un sottile strato connettivo (l'endomisio), vengono raggruppate da un altro tessuto di natura connettivale (il perimisio) a creare i fascicoli. Più fascivoli, circondati da altro tessuto connettivale, vengono raggruppati a loro volta a costituire l'intero muscolo, rivestito da una guaina chiamata epimisio.

Il diametro della fibra muscolare varia da 20 a circa 100 micron e la sua lunghezza da pochi millimetri fino a più di 10 cm. É rivestita da una membrana, il sarcolemma, e al suo interno sono stipate migliaia di miofibrille, che decorrono parallele tra loro. La miofibrilla è un sottilissimo cilindro del diametro di 1 - 3 micron la cui striatura, caratteristica del muscolo striato, è dovuta alla regolare alternanza nelle miofibrille, di dischi scuri e chiari. Il segmento di miofibrilla, comprendente una banda scura e due mezze bande chiare, è detto sarcomero e rappresenta l'unità contrattile del muscolo scheletrico.

Il sarcomero è costituito da fasci di filamenti, regolarmente intercalati tra loro, i miofilamenti. Essì si distinguono, in base al loro spessore, in filamenti spessi, formati ciascuno dall'intreccio di diverse centinaia di molecole di miosina, e sottili, formati da actina ed altre molecole.

Un muscolo di una data lunghezza è dunque formato da numerose fibre muscolari poste in parallelo, ciascuna composta da numerose miofibrille, poste in parallelo e di lunghezza uguale a quella del muscolo. Ciascuna miofibrilla è formata da un certo numero di sarcomeri, di lunghezza uguale tra loro e posti in serie. Quando una singola miofibrilla si contrae sviluppa una tensione uguale a quella unitaria di un singolo sarcomero. Il muscolo, contraendosi in toto, produce una forza complessiva data dalla somma delle forze sviluppate da ogni singolo elemento posto in parallelo. In definitiva, la forza prodotta da un muscolo con la contrazione, è proporzionale al numero di miofibrille poste in parallelo, cioè alla sua sezione trasversa.

L'accorciamento totale è dato dalla somma degli accorciamenti dei singoli sarcomeri, pertanto è proporzionale al numero di sarcomeri posti in serie, cioè alla sua lunghezza.



Figura 3.1: A. Muscolo connesso all'osso attraverso il tendine. B. All'interno del muscolo le fibre vengono raggruppate in fascicoli. C. Ciascuna fibra contiene filamenti di miofibrille che corrono lungo tutta la lunghezza del muscolo. D. L'unità contrattile, il sarcomero. Numerosi sarcomeri sono connessi in serie lungo ciascuna miofibrilla. L'accorciamento muscolare (contrazione) avviene nel sarcomero quando i miofilamenti di actina e miosina all'interno del sarcomero scivolano l'uno sull'altro.

I muscoli scheletrici sono innervati da grosse fibre nervose, le cui origini sono i motoneuroni alfa, situati nelle corna anteriori del midollo spinale. Tali nervi si portano al muscolo tramite i loro prolungamenti assonali (fibra nervosa), le cui ramificazioni terminali prendono ciascuna contatto con una singola fibra muscolare.

L'unità funzionale del muscolo è l'unità motoria (UM): è l'insieme del motoneurone e delle fibre muscolari che esso innerva. Le fibre muscolari di una unità motoria hanno nel muscolo una disposizione casuale, non contigua, sono, cioè, frammiste a mosaico con fibre muscolari appartenenti ad altre unità motorie. Le unità motorie variano molto per quanto riguarda il numero di fibre muscolari: vi sono UM con 10 - 20 fibre muscolari, altre con più di 2000 e ciò influenza la forza generata dall'attività motoria stessa. Generalmente i muscoli preposti al controllo dei movimenti più fini hanno un basso numero di fibre muscolari per unità motoria; invece, i muscoli preposti a movimenti più grossolani sono usualmente caratterizzati da un elevato numero di fibre muscolari per unità motoria.

La contrazione muscolare

Si possono distinguere tre tipi di contrazione muscolare, in base alla resistenza che l'ambiente oppone al moto:

- Contrazione concentrica, è la contrazione nella quale il muscolo si accorcia e la forza generata è sufficiente a vincere la resistenza. Le due inserzioni del muscolo si avvicinano tra loro;
- Contrazione eccentrica, la forza generata è insufficiente a vincere la resistenza (o il carico esterno) e le fibre muscolari, pur contraendosi, si allungano. La contrazione eccentrica è usata per decelerare una parte del corpo o un oggetto, oppure per abbassare un carico dolcemente, al posto di lasciarlo cadere. Le due inserzioni del muscolo si allontanano l'una dall'altra;
- Contrazione isometrica , in essa, il muscolo è attivato, ma invece di permettergli di accorciarsi o allungarsi, viene tenuto ad una lunghezza costante. Un esempio si ha quando, tenendo un oggetto in mano di fronte a noi, il peso dell'oggetto tende a far abbassare l'avambraccio, ma quest'ultimo e la mano si oppongono al moto con una forza uguale e contraria. Dato che il braccio non solleva né abbassa l'oggetto, il muscolo bicipite si sta contraendo isometricamente.

Per capire come avviene la contrazione muscolare bisogna introdurre il concetto di potenziale d'azione. La cellula e il suo ambiente sono assimilabili a due soluzioni ricche di ioni, separate da una membrana semipermeabile. La membrana cellulare è selettiva in quanto i suoi pori hanno dimensione tale da da lasciar passare liberamente gli ioni Cl-, K+, ostacolare quelli Na+ ed impedire il passaggio degli anioni organici A-. La distribuzione degli ioni determina una differenza di potenziale elettrico ai lati della membrana come mostrato in figura 3.2. Questo potenziale di membrana a riposo ha un valore costante, compreso tra -70 e -90 mV e la sua variazione viene chiamata potenziale d'azione. Quando varia la tensione al colle dell'assone per effetto delle sinapsi eccitratrici agenti sul neurone, si ha una variazione della permeabilità allo ione sodio nello stato di eccitazione o una diminuzione di

questa che al contrario porta ad un aumento della permeabilità allo ione potassio nel ritorno allo stato di riposo. Il potenziale d'azione è dunque alla base del fenomeno di eccitabilità delle cellule, in particolare quelle nervose e muscolari, che ne fanno uso rispettivamente per trasmettere messaggi e per rispondere agli stessi. Il meccanismo di variazione della permeabilità della membrana e la relativa variazione di potenziale è detto depolarizzazione e si propaga lungo tutta la fibra nervosa, in quanto determina un movimento di ioni dalla regione contigua non stimolata verso quella stimolata, con depolarizzazione della regione non stimolata, che a sua volta stimola la regione adiacente e così via. Lo stimolo quindi si propaga ad una fibra (cellula) muscolare con la quale la fibra nervosa è in rapporto. Solo il superamento di una soglia (dell'ordine di 30 mV) provocherà il trasferimento dell'informazione e la sua propagazione lungo la fibra stessa, senza variazione della forma d'onda ed a a una velocità praticamente costante (40-70 m/s per le fibre nervose e 1, 3 - 4, 7 m/s per le fibre muscolari). Il complesso meccanismo chimico che viene indotto dalla depolarizzazione delle membrane cellulari porta all'effetto di contrazione delle fibre muscolari, a causa dello scivolamento dei filamenti di miosina e actina e dunque di tutti i sarcomeri che formano la lunga fibra muscolare.



Figura 3.2: Fisiologia della membrana e potenziale d'azione.

La graduazione della contrazione muscolare è ottenuta grazie a due diversi meccanismi:

1. reclutamento di nuove unità motorie all'aumentare dello sforzo (sommazione spaziale); 2. aumento della frequenza di scarica della singola unità motoria (sommazione temporale).

La stimolazione della fibra muscolare, poco dopo che è stata evocata una precedente contrazione, dà luogo ad una seconda risposta contrattile che si somma alla precedente. Se, infatti, il secondo stimolo arriva alla fibra muscolare prima che la tensione del primo sia decaduta, la seconda contrazione si sommerà alla prima, dando luogo ad un significativo aumento del picco di tensione.

3.2.2 Il segnale emg superficiale

In condizioni normali, un potenziale d'azione si propaga lungo il motoneurone e attiva tutti i suoi rami, questi a loro volta, attivano tutte le fibre muscolari dell'unità motoria. Quando la membrana post-sinaptica della fibra muscolare è depolarizzata, tale depolarizzazione si propaga in entrambe le direzioni. La membrana depolarizzata, che è accompagnata da un movimento di ioni, genera un campo elettromagnetico nelle vicinanze delle fibre muscolari. Un elettrodo localizzato in tale campo rileverà il potenziale, la cui escursione temporale è nota come potenziale d'azione, la cui ampiezza dipende dal diametro della fibra muscolare, dalla distanza tra la fibra muscolare e la posizione dell'elettrodo e dalle proprietà filtranti dello stesso.

Il segnale risultante, presente nel punto di rilevazione, somma spazio temporale dei singoli potenziali d'azione prodotti dalle depolarizzazioni delle fibre muscolari di una unità motoria, è chiamato MUAP (motor unit action potential). La disparità nel tempo di attivazione per fibre appartenenti ad una stessa unità motoria, dovuta a vari fattori chimico-meccanici porta a processi di scarica di tipo casuale.

Nella zona di rilevazione degli elettrodi, ci sono i contributi di altre unità motorie, quindi si rilevano una serie di MUAPs. Questi possono avere ampiezza e forma simile, se ogni fibra muscolare appartenente alla rispettiva unità motoria ha la stessa distanza dalla zona di rilevamento. A seconda dell'elettrodo usato, si riscontrano variazioni sulla forma, la fase e la durata dei MUAPs. L'ampiezza e la forma di un MUAP osservato sono, pertanto, funzione della disposizione geometrica dell'unità motoria, del tessuto muscolare e delle proprietà degli elettrodi utilizzati.

Con la contrazione muscolare prolungata nel tempo, tutte queste componenti vengono a sommarsi in maniera irregolare, creando il cosiddetto tracciato di interferenza sulla superficie cutanea, un segnale elettrico altresì noto come segnale elettromiografico superficiale.



Figura 3.3: Misura intramuscolare di un MUAP. h(t): risposta ad un singolo impulso neurale.

Il segnale elettromiografico di superficie è un segnale aleatorio e la sua misura è una realizzazione di un processo stocastico non stazionario a media nulla. La stazionarietà richiede che le proprietà statistiche non dipendano da un riferimento temporale assoluto, cosa che nel caso del segnale elettromiografico non vale, per una serie di motivi che verranno spiegati nelle prossime sezioni, tra i quali il più significativo è la fatica muscolare, che porta a variazioni dello spettro del segnale. La particolarità di questo genere di segnali impone il fatto che vengano inquadrati nell'ambito dei segnali aleatori e dunque che vengano studiati con le relative techiche. Per analizzare un processo stocastico si utilizzano alcune funzioni fondamentali quali media, varianza, autocorrelazione, densità spettrale di potenza e altre features che verranno descritte nel dettaglio nelle prossime sezioni di questo capitolo.

Caratteristiche elettriche del segnale EMG

Lo spettro del segnale EMG è composto essenzialmente in due parti:

- 1. una dovuta alle frequenze di sparo delle UM più grandi e reclutate per ultime, che contribuiscono con picchi nello spettro, in bassa frequenza, attorno alla frequenza media di sparo delle UM, tra i 10-40 Hz;
- 2. una seconda parte dovuta alla forma dei MUAP, e va dai 50 fino a oltre i 100 Hz, in funzione di vari fattori, quali la forma e la disposizione degli elettrodi, la distanza delle fibre, la fatica muscolare, ed altri fattori.

Lo spettro complessivo del segnale, per la complessità e la rapidità delle reazioni chimiche nel reclutamento delle UM arriva fino a circa 2 KHz, ma la potenza del segnale ha la sua massima concentrazione nel range 30-150 Hz.

Le caratteristiche frequenziali del segnale elettromiografico inoltre non risultano costanti e son dipendenti da alcuni parametri fisiologici, tra i quali:

- 1. Livello di tensione muscolare e fatica muscolare;
- 2. Lunghezza delle fibre muscolari;

Numerosi studi hanno mostrato evidenti variazioni dello spettro EMG con l'aumentare della fatica muscolare, in particolare uno spostamento verso le basse frequenze dello spettro col procedere dell'esercizio. In corrispondenza si nota un'incremento dell'ampiezza. Altro parametro fondamentale è la lunghezza delle fibre muscolari, per le quali è stata individuata una corrispondenza inversa tra la loro lunghezza e la frequenza media dello spettro del segnale elettromiografico. Un allungamento delle fibre corrisponde perciò ad una diminuzione della frequenza media. Un altro altro aspetto del fenomeno elettromiografico, è la variazione delle sue caratteristiche con il variare dell'età, che rende difficilmente comparabile il segnale elettromiografico tra soggetti diversi [38].

L'ampiezza del segnale varia in base al tono muscolare ed alla sezione traversa del muscolo preso in considerazione, ma in generale in un range di $\pm 10mV$.

Il segnale EMG del muscolo a riposo presenta una componente basale che dipende dai rumori elettrici sulla superficie cutanea e dal tono muscolare, tipicamente questo in valore assoluto medio non è superiore a 3-5 μV assumendo una adeguata preparazione della superficie cutanea e la misura del segnale con un amplificatore allo stato dell'arte [39].

Campi di utilizzo del segnale EMG

La principale particolarità che rende i segnali elettromiografici davvero importanti in molti campi di ricerca è quella di poter osservare il comportamento dei muscoli dall' "interno", con l'utilizzo di tecniche non invasive (nel caso degli elettrodi superficiali), permettendo di valutare la funzionalità neuromuscolare in termini sia qualitativi che quantitativi.

In campo medico l'esame elettromiografico fornisce risultati caratteristici che permettono di individuare e distinguere malattie muscolari (miopatie) come ad esempio le distrofie, o patologie al sistema nervoso periferico (neuropatie) come neuriti e sindrome del tunnel carpale, o applicazioni rivolte allo studio e terapia del dolore, analisi di postura del corpo umano e ortopedia. In



Figura 3.4: Spettro in frequenza del segnale sEMG.

campo riabilitativo, grazie all'elettromiografia, possono essere esseguiti studi pre/post-operatori o monitoraggi dell'efficacia delle terapie di riabilitazione. Un uso attivo dei segnali elettromiografici viene fatto per il comando di protesi e sistemi robotici riabilitativi, che permettono inoltre di ottenere ottimi risultati nella riabilitazione neurologica.

In campo sportivo ad esempio una delle applicazioni principali consiste nel verificare e migliorare con allenamenti "ad hoc" tutte le sinergie tra i muscoli impiegati nei movimenti, in modo tale da migliorare le prestazioni dell'atleta, o di studiare l'affaticamento muscolare e le sue cause.

Di recente utilizzo è nel campo dell'ergonomia, permettendo lo studio della prevenzione dei rischi, l'analisi della domanda, della progettazione ergonomica o la certificazione di prodotti di uso commerciale.

Relazione EMG-forza muscolare

Esiste un certo consenso nelle opinioni della comunità scientifica secondo il quale il legame tra emg e forza sia relativamente lineare in condizioni isometriche, per il quale dato un incremento del segnale elettromiografico si ha in concomitanza un aumento della forza muscolare sviluppata, prodotta da una combinazione dell'aumento delle unità motorie reclutate e l'aumento della frequenza di sparo nelle unità motorie. Tuttavia son state suggerite diverse relazioni tra le due entità in gioco, includendo sia quelle lineari che quelle curvilinee per differenti muscoli [40].

Nei casi di contrazioni concentriche o eccentriche, le relazioni descritte tra emg e forza sono piuttosto controverse. In letteratura la maggioranza degli studi hanno focalizzato la loro attenzione sui casi isochinetici, pochi studi hanno cercato di spiegare la relazione durante movimenti non vincolati, ed in generale le relazioni identificate son sempre di tipo non lineare [41]. É importante introdurre il fatto che Il segnale emg acquisito su una parte del muscolo non è necessariamente rappresentativo del comportamento del muscolo intero.

Aspetto fondamentale della relazione tra il segnale elettromiografico e la forza muscolare è il ritardo elettromeccanico, in quanto la depolarizzazione delle membrane muscolari avvengono prima dell'evento meccanico della forza. Il ritardo è dovuto a diversi fattori quali la propagazione dei potenziali d'azione, il rilascio del calcio dal reticolo sarcoplasmatico, il movimento dei ponti di actina-miosina, l'elasticità degli elementi meccanici del muscolo messi in moto, principalmente i filamenti di miosina ed i tendini. Un'ampia variabilità nelle sue stime è stata riportata in letteratura in un range compreso tra i 30-150 ms ed è dipendente dal livello di forza, dalla lunghezza delle fibre, il tipo di contrazione (isometrica, concentrica, eccentrica) e tono muscolare [42].



Figura 3.5: Segnale Elettromiografico e ritardo elettromeccanico della forza muscolare.

3.3 Strumenti e metodologie di misura

3.3.1 Elettrodi e preparazione della cute

Le sorgenti del segnale elettromiografico sono separate dagli elettrodi di registrazione da tessuto biologico, principalmente la pelle e lo strato di grasso sottocutaneo. Questi due elementi si comportano come un filtro passa-basso nella distribuzione spaziale del potenziale, ma il loro effetto è differente in base alle metodologie di misura ed al tipo di elettrodi utilizzati.

La misura del segnale EMG, come già accennato, è di due tipi: una intramuscolare e dunque invasiva, ed una superficiale. La prima è prevalentemente utilizzata per la ricerca e la diagnosi in campo medico ed in questo caso l'effetto dei tessuti è ridotto in quanto gli elettrodi sono posti molto vicini alla sorgente muscolare. Gli elettrodi utilizzati sono costituiti da un sottilissimo e flessibile cavo metallico, affondato nel ventre muscolare tramite una sottilissima cannula di metallo, in modo tale che l'invasività del contatto sia la minima possibile.

L'elettromiografia di superficie invece viene eseguita tramite l'utilizzo di elettrodi metallici o di elettrodi all'argento/cloruro d'argento. Tra la superficie metallica dell'elettrodo e la cute, è sempre necessario interporre uno strato di pasta adesiva conduttiva, in modo tale da massimizzare la superficie di contatto dell'elettrodo stesso. La condizione ideale per misurare la distribuzione potenziale sulla pelle è avere un elettrodo puntiforme connesso ad un voltmetro con impedenza di ingresso infinita che misura il voltaggio rispetto ad un punto a potenziale nullo. Nella situazione reale bisogna fare alcune considerazioni:

- la pelle è un tessuto sufficientemente conduttivo, costituita da cellule il cui materiale interno ed esterno è costituito da soluzioni con ioni la cui corrente è portata da elettroni, perciò l'interfaccia è intrinsecamente rumorosa;
- l'interfaccia elettrodo-pelle è molto complessa in quanto l'impedenza è principalmente capacitiva/resistiva piuttosto dipendente dalla corrente e dalla frequenza, ed inoltre incorpora un generatore di corrente che tiene conto dell'interfaccia elettrolita-metallo;
- l'elettrodo metallico e la pasta conduttiva impongono l'equipotenzialità sulla superficie della misura, e questo modifica la distribuzione di potenziale attorno all'area di misura, considerazione importante se si vuole posizionare l'elettrodo di riferimento a breve distanza dal punto di misura
- l'impedenza dell'amplificatore stessa per quanto elevata, è dipendente dalla frequenza.

L'elettrodo reale si comporta a tutti gli effetti come un filtro spaziale/frequenziale, in quanto media i contenuti di ciascuna fibra muscolare nello spazio (filtro spaziale) e allo stesso tempo filtra il contenuto frequenziale del segnale in quanto è un sistema intrinsecamente passa-basso. La frequenza di taglio degli elettrodi è inversamente proporzionale al suo diametro, perciò la loro dimensione dovrà essere accuratamente decisa. Diminuendo l'area dell'elettrodo si aumenta l'impedenza dell'interfaccia elettrodo-cute. Inoltre, nel punto di contatto c'è rumore dovuto a fenomeni che danno luogo ad interferenza, il rumore diminuisce all'aumentare dell'area dell'elettrodo. Con l'aumentare dell'area dell'elettrodo diminuisce inoltre anche la selettività sulla misura muscolare, portando all'aumento di fenomeni di crosstalk muscolari di cui si parlerà in seguito. La dimensione (e l'eventualmente la forma) dovrà essere studiata anche in base alla dimensione del muscolo stesso. Tipicamente il diametro degli elettrodi di misura va dai 5-10/15 mm, mentre gli elettrodi di terra/riferimento possono avere dimensioni anche più grandi.



Figura 3.6: Confronto tra la misura intramuscolare del segnale emg (a) e la misura superficiale (b).

Preparazione della cute

La qualità della misura dei segnali EMG è fortemente dipendente dalla preparazione della pelle oltre che dal posizionamento stesso degli elettrodi. Il contatto elettrodo-cute necessita stabilità è spesso necessario depilare, pulire e strofinare la pelle con un panno leggermente abrasivo e bagnato con acqua o un solvente. Il gel conduttivo aiuto a migliorare e stabilizzare il contatto ed a diminuire l'impedenza della pelle. I moderni sistemi di acquisizione elettromiografici necessitano che il livello d'impedenza della pelle stia nel range 5-50 $K\Omega$ tra la coppia di elettrodi usati per la misura. Quando la cute assume una colorazione rosacea accesa tendente al rosso, allora è da considerarsi pronta per l'applicazione dell'elettrodo.

Posizionamento degli elettrodi

Il posizionamento degli elettrodi influisce pesantemente sulla qualità della misura elettromiografica e sulla selettività per quanto riguarda l'origine muscolare del segnale EMG. Alcuni progetti di ricerca si son prefissi di standardizzare sia le metodologie che i punti di origine per poter avere misure facilmente comparabili e significative. Tra questi il progetto europeo SENIAM (Surface ElectroMyoGraphy for the Non-Invasive Assessment of Muscles) propone delle raccomandazioni sulle proprietà e procedure di posizionamento degli elettrodi e altri sensori per l'acquisizione del segnale EMG. Per l'applicazione degli elettrodi di riferimento le raccomandazioni considerano un sistema di riferimento anatomico, basato sulle più importanti superfici e prominenze ossee o altre strutture che possono essere facilmente individuate e tastate. L'uso di questi *punti di repere* (figura 3.7) serve a localizzare la posizione dell'area scelta per l'applicazione dell'elettrodo di riferimento.

Gli elettrodi devono essere posizionati lungo il decorso dele fibre muscolari, paralleli alla direzione della contrazione del muscolo. É importante che l'elettrodo rimanga nella zona della massa muscolare, e dunque sia ben fissato e stabile, così che durante la contrazione si eviti la migrazione dell'elettrodo durante il movimento. Oltre a questo è importante posizionare l'elettrodo in prossimità della massima sezione del ventre muscolare, in modo da massimizzarne l'ampiezza massima ed evitare inoltre che si perdano informazioni elettromiografiche a causa della migrazione del ventre muscolare stesso durante la contrazione (figura 3.8).

3.3.2 Amplificatore

Come detto nelle sezioni precedenti, il segnale elettromiografico ha ampiezze massime che possono raggiungere i 10 mV in base al gruppo muscolare, e per essere trattato e processato è necessario poterlo amplificare prima di filtrarlo e campionarlo. I possibili metodi di misura dei segnali elettromiografici sono di due tipi, una misura monopolare ed una differenziale. Nel primo caso per ciascun muscolo è necessario un solo elettrodo oltre a quello di riferimento e ciò comporta una selettività migliore, ma un peggioramento per quanto riguarda la presenza del rumore che sporca il segnale. La soluzione differenziale ha bisogno dell'utilizzo di due elettrodi, è tipicamente meno selettiva perchè i due elettrodi devono essere posti ad una distanza che non superi i



Figura 3.7: Posizionamento degli elettrodi di misura sul ventre muscolare.



Figura 3.8: Posizionamento degli elettrodi di misura sul ventre muscolare.

2-3 cm tra i loro centri geometrici, portando ad occupare complessivamente un'area più grande rispetto al caso monopolare. Il notevole vantaggio di questa soluzione si ha nel fatto che la misura differenziale permette di ridurre (se non addirittura annullare) il rumore indotto sulla pelle da parte dell'ambiente esterno, ed il rumore che viene indotto sui cavi di trasmissione del segnale. Per eliminare il cosiddetto modo comune tra i due singoli segnali, l'amplificatore deve avere un Rapporto di reiezione in modo comune (CMRR) superiore a 95 dB. Per ridurre la presenza del rumore, in molti casi è necessario poter amplificare il segnale prima possibile nella dimensione spaziale, per evitare tutti i rumori indotti nei cavi di trasmissione. Questo è possibile tramite elettrodi attivi, dotati di preamplificatore differenziale, che permettono da subito di eliminare tutti i possibili rumori sul lato cutaneo



Figura 3.9: Posizioni anatomiche degli elettrodi - lato frontale. A sinistra i punti per gli elettrodi intramuscolari, a destra gli elettrodi di superficie.

e dunque di non amplificarli. Oltre a questo, gli stessi rumori di trasmissione che vengono a sommarsi son minori (in rapporto) rispetto al segnale EMG preamplificato, permettendo di ottenere un rapporto segnale rumore migliore rispetto ai normali elettrodi passivi. Tipicamente l'amplificatore ha guardagni di amplificazione tra 500-1000 (elettrodi attivi-elettrodi passivi), così da poter ottenere un segnale di buona ampiezza prima di essere campionato. L'impedenza d'ingresso dell'amplificatore generalmente devono essere almeno dieci volte l'impedenza data dagli elettrodi.

3.3.3 Artefatti e disturbi

La qualità del segnale elettromiografico, a causa del suo basso livello di ampiezza elettrica a livello della cute, dipende pesantamente da alcuni fattori di disturbo, alcuni di questi esterni, ed altri intrinseci al segnale EMG stesso.

I rumori esterni più comuni in questo genere di misure son legati alla presenza di altri campi elettromagnetici all'interno del locale di misura, come l'alimentazione elettrica e la presenza di apparecchiature che generano il disturbo, come motori elettrici, alimentatori e quant'altro. É facile pensare che la cute umana si comporti come un'antenna capace di subire l'induzione che questi disturbi esterni provocano su di essa, generando potenziali sulla pelle e correnti parassite di vario genere. Tipicamente i disturbi di tipo elettrico (dell'alimentazione) si stabiliscono sulla frequenza di alimentazione stessa, ovvero 50 Hz nel caso dell'Italia. Diverso è il caso dei disturbi di tipo magne-



Figura 3.10: Posizioni anatomiche degli elettrodi - lato dorsale. A sinistra i punti per gli elettrodi intramuscolari, a destra gli elettrodi di superficie.

tico (dovuti alle correnti) come nel caso di azionamenti elettrici di motori, in presenza di elettronica di potenza. Solitamente questo genere di disturbi vengono schermati grazie ad un buon impianto di messa a terra, e nel caso della misura differenziale del segnale EMG di solito vengono abbattuti senza troppa difficoltà. La loro frequenza è di solito centrata attorno alla frequenza portante dell'elettronica di potenza. Il classico esempio è l'alimentazione in PWM dei motori in continua, che di solito dovrebbe superare i 20 KHz sia per un semplice filtraggio elettrico di tali disturbi che l'eliminazione dei rumori acustici delle macchine.

Nel caso dei disturbi intrinseci al segnale EMG, il caso è diverso, in quanto dipendono da fattori muscolari o meccanici.

Crosstalk muscolare e disturbi esterni

In base alla zona anatomica che si intende monitorare tramite l'elettromiografia è più o meno presente un fenomeno detto di crosstalk muscolare. Basti pensare ai muscoli che interessano l'articolazione della spalla, nella cui zona son concentrati numerosi gruppi muscolari di cui molti di stabilizzazione dell'articolazione stessa, ed altri più interessati al movimento. La misura elettromiografica di un gruppo muscolare, inevitabilmente verrà disturbata dalla presenza dei MUAPs degli altri gruppi muscolari presenti nelle vicinanze in base al livello di contrazione. Il fenomeno può essere piuttosto visibile soprattuto nel caso a riposo dei gruppi muscolari di movimento, il cui livello basale di segnale elettromiografico è disturbato dalla contrazione di stabilizzazione degli altri muscoli presenti nell'articolazione. In genere i fenomeni di crosstalk muscolare non superano il 10-15 % del contenuto del segnale.

Il caso più importante ed evidente di crosstalk muscolare è quello dell'ecg cardiaco che va ad influenzare tutti i muscoli del torace e della spalla. Se ne parlerà nel dettaglio nelle prossime sezioni.

Artefatti da movimento

L'importanza di avere una bassa impedenza di contatto degli elettrodi e della loro stabilità meccanica, è evidente quando nel segnale elettromiografico si intravedono i cosiddetti artefatti da movimento. Il movimento dell'elettrodo porta dunque ad un disturbo per insufficienza (o mancanza) di contatto che di solito a livello elettrico è presente tra nel range tra i 0-10 Hz. Questo genere di disturbo può essere evitato con una buona pulizia della pelle, con un fissaggio stabile degli elettrodi e dei cavi di trasmissione tramite nastro adesivo. É importante evitare il più possibile l'utilizzo di fasce che possano ostacolare lo scorrimento del ventre muscolare durante il movimento.



Figura 3.11: Segnale EMG con artefatti di movimento dopo ogni contrazione.

3.4 Trattamento dei segnali ed estrazione delle features

3.4.1 Campionamento e filtraggio segnali

La catena di trattamento del segnale EMG è composta innanzitutto dall'amplificatore differenziale ad alto gain, in modo tale da migliorare il più possibile il rapporto segnale-rumore. Il segnale così amplificato ha bisogno di essere filtrato prima di essere inviato alla scheda di acquisizione. Come detto in precedenza, il segnale EMG ha uno spettro che arriva fino a 2 KHz, ma con contenuto significativo in potenza concentrato tra i 30 e 150 Hz. Il SENIAM consiglia di campionare il segnale con frequenze di campionamento dell'ordine dei 2 KHz, e filtrare passa basso intorno alla frequenza di Nyquist di 1 KHz.

3.4.2 Estrazione features dal segnale emg

Come già detto nelle sezioni precedenti, la misura del segnale elettromiografico è una realizzazione di un processo stocastico, e dunque per questo motivo lo studio delle sue caratteristiche è da inquadrare nella branca dei segnali aleatori. L'utilizzo diretto di questi segnali per il controllo o la classificazione dei movimenti risulta dunque inutile, ed ha bisogno di ulteriori passaggi (oltre al filtraggio) per poter essere efficace. L'estrazione delle features è un metodo per estrarre informazioni "nascoste" ed utili dai segnali EMG, per poter dunque separare le informazioni utili da quelle non necessarie, e per rimuovere tutte le possibili interferenze. Alcune features, rispetto ad altre, sono robuste per diversi tipi di rumore e dunque permettono di evitare un intenso pre-processing del segnale. Una buona feature deve rispondere a tre requisiti fondamentali:

- massima separabilità tra le classi;
- robustezza;
- complessità.

Prima della scelta del set di features è importante inquadrare bene il tipo di problema che si vuole risolvere. I problemi di classificazione hanno esigenze diverse rispetto ad un problema di identificazione. A tal scopo è importante capire quale rappresentazione del segnale è più efficace rispetto ad altre. I modi di rappresentare un segnale sono tipicamente di tre tipi:

• dominio del tempo;

- dominio della frequenza;
- dominio tempo-frequenza.

L'informazione che le features temporali possono dare è estraibile solitamente in maniera piuttosto rapida, e con algoritmi che non richiedono grossi tempi di calcolo. La complessità computazionale aumenta nel caso delle features frequenziali ed ancora di più per le features nel dominio misto tempo-frequenza come le trasformate wavelet e simili. È importante quindi scegliere delle features che abbiano un ottimo rapporto di "rapidità"/densità di informazione. La questione è evidente nel caso di controllo di sistemi che hanno bisogno di un alto grado di real-time, nei quali diventa davvero critica la scelta delle features. Altro elemento importante nella preparazione del set di features è la loro quantità. Una delle prime domande che bisogna porsi è quante features saranno sufficienti per descrivere in maniera "ottima" il pro*blema?* Rispondere a questa domanda non è sempre facile, ed infatti la prassi è quella di selezionare un numero di features "abbondanti" e di eseguire infine una riduzione del loro spazio, tramite analisi di correlazione come la PCA (metodo delle componenti principali). Ma anche a tal proposito è importante selezionare un numero ed un tipo di features che non siano troppo ridondanti, cosa che peggiorerebbe la qualità della classificazione o dell'identificazione dei parametri.

Le features nel dominio spettrale o in quello della frequenza sono usate principalmente per gli studi sulla fatica e per l'analisi del reclutamento delle unità motorie. Le principali derivano dalla densità di potenza spettrale, definita spesso come la trasformata di Fourier della funzione di autocorrelazione del segnale elettromiografico. Può essere stimata anche tramite il periodogramma o altri metodi parametrici come il modello auto-regressivo.

Le features nel dominio misto di tempo-frequenza (come la trasformata wavelet) hanno il difetto che per essere calcolate necessitano di tempi di calcolo piuttosto lunghi, che le rendono difficili da utilizzare in casi in cui l'applicazione sia un controllo di tipo real-time. Si sono rivelate utili nel caso in cui si abbia a che fare con segnali molto disturbati dal tracciato ECG del cuore, o di altri disturbi ripetitivi identificabili nella forma con dei template.

Features nel dominio del tempo

Le features temporali hanno come punto di forza la loro caratteristica principale, ovvero la semplice implementazione e la rapidità di calcolo, in quanto non necessitano trasformazioni e sono calcolate direttamente dalla serie temporale del segnale EMG. Gli svantaggi principali sono essenzialmente due. Il primo è legato alla non-stazionarietà delle caratteristiche del segnale, ed il secondo è legato al fatto che il loro calcolo è basato essenzialmente sui valori d'ampiezza e perciò risentono di più dei fenomeni di crosstalk muscolare e altri rumori. Le principali features nel dominio del tempo sono riportate in tabella 3.1

Feature	Formulazione	Caratteristiche
Integrated EMG	$\text{IEMG} = \sum_{i=1}^{N} x_i $	Integrale del valore assoluto della se- rie numerica EMG per una finestra ampia N campioni, correlato alla se- quenza di sparo delle unità motorie, utilizzato di solito per il rilevamento dell'inizio del movimento
Mean absolute va- lue	$MAV = \frac{1}{N} \sum_{i=1}^{N} x_i $	Una delle features più usate, è la media di una finestra del segnale raddrizzato
Weighted Mean absolute value	$MAV_w = \frac{1}{N} \sum_{i=1}^N w_i x_i $	Simile alla feature precedente, ma col vantaggio di poter dare un pe- so diverso ai campioni della fine- stra temporale ampia N. Utile per fare smoothing di segnale e per aumentare la robustezza del MAV
Simple square in- tegral	$SSI = \sum_{i=1}^{N} x_i^2$	É l'energia di una finestra di segnale elettromiografico
Variance of EMG	$VAR = \frac{1}{N-1} \sum_{i=1}^{N} x_i^2$	La varianza è un altro indice legato alla potenza del segnale stesso. In generale la varianza è definita come la media dei quadrati della deviazio- ne dal valore medio, in questo caso il valore medio del segnale EMG è nulla.
Root mean square	$RMS = \sqrt{\sum_{i=1}^{N} x_i^2}$	É modellato come l'ampiezza mo- dulata di un processo casuale gaus- siano rispetto ad una forza costante ed una contrazione non affaticante. Simile al metodo della deviazione standard.

Tabella 3.1: Principali features nel dominio del tempo

Tabella 3.1: continua nella prossima pagina

Feature	Formulazione	Caratteristiche
Waveform lenght	WL = $\sum_{i=1}^{N-1} x_{i+1} - x_i $	É la misura della complessità del se- gnale EMG, definita come la lun- gheezza comulativa della sua forma d'onda
Average amplitu- de change	AAC = $\frac{1}{N} \sum_{i=1}^{N-1} x_{i+1} - x_i $	Equivalente alla WL tranne per il fatto che è mediata
Zero Crossing	$ZC = \sum_{i=1}^{N-1} (sgn(x_i \times x_{i+1}))$ $\cap (x_{i+1} - x_i \ge \text{soglia}))$	É il conteggio del numero di volte che il segnale attraversa lo zero. Al- tamente correlato con la frequen- za media/mediana della FFT del segnale. Per evitare il più possi- bile l'influenza del rumore, l'attra- versamento deve superare una certa soglia.
Auto-regressive coefficients	$x_i = \sum_{p=1}^p a_p x_{i-p} + w_i$	Il modello auto-regressivo è un mo- dello di predizione che descrive cia- scun campione del segnale come combinazione lineare dei campioni precedenti più un termine di peso dell'errore. I coefficienti a_p stes- si vengono usati come feature del segnale.

Tabella 3.1: continua dalla pagina precedente

Tabella 3.1: si conclude dalla pagina precedente

Capitolo 4

Strumenti e Metodi

In questo capitolo vengono descritti gli strumenti ed i metodi utilizzati durante il lavoro di tesi, focalizzando l'attenzione sull'esoscheletro leggero L-Exos sviluppato nel laboratorio PERCRO dell'istituto TeCIP della Scuola Superiore Sant'Anna, le apparecchiature di misura per l'acquisizione dei segnali elettromiografici e le specifiche utilizzate per il campionamento ed il filtraggio dei segnali. In particolare verrà descritto il metodo di rimozione del disturbo di crosstalk ECG dai gruppi muscolari del tronco.

4.1 L'esoscheletro leggero L-Exos

4.1.1 Descrizione

L-Exos (Light Exoskeleton) è un'interfaccia aptica esoscheletrica seriale per l'arto superiore umano, sviluppata nei laboratori del PERCRO Scuola Superiore S.Anna, capace di fornire una forza controllabile al centro del palmo della mano destra, orientata lungo qualsiasi direzione nello spazio. L'esoscheletro ha 5 DOF ed ha dimensioni pari ad un arto superiore umano medio. Il dispositivo è adatto per applicazioni in cui son richiesti sia il tracking del movimento che una retroazione in forza, come ad esempio l'interazione umana con ambienti in realtà virtuale o compiti di teleoperazione. La configurazione base del L'Exos prevede una maniglia montata nell'ultimo link, ma son previste altre configurazioni nelle quali la maniglia è sostituita da un esoscheletro per la riabilitazione della mano, in particolare rivolta alle funzioni di presa con il palmo della mano.



Figura 4.1: L'esoscheletro leggero L-Exos.

4.1.2 Cinematica diretta

L'esoscheletro L-Exos è caratterizzato da una cinematica seriale che consiste in 5 giunti rotoidali (vedi figura 4.2), dei quali i primi 4 sono attuati e sensorizzati mentre l'ultimo è solo sensorizzato. I primi tre assi di rotazione sono incidenti e mutuamente ortogonali due a due, questo per emulare la cinematica di un giunto sferico con lo stesso centro di rotazione della spalla umana, supposta in posizione fissa nello spazio. L'workspace per il giunto della spalla umana è assunto perciò come un settore sferico come mostrato in figura 4.3. Tramite processo di ottimizzazione, l'orientazione del primo



Figura 4.2: Cinematica dell'esoscheletro L-Exos.

asse è stata scelta in modo da massimizzare l'workspace del giunto di spalla,

evitando singolarità ed eventuali interferenze tra il manipolatore e l'utente oltre alla definizione della forma dei link stessi. Come risultato dell'analisi cinematica, l'orientazione finale del primo asse (quello fisso) è stata scelta per essere obliqua rispetto sia al piano orizzontale che a quello verticale, mentre il terzo asse è stato concepito per essere coincidente con l'asse ideale dell'arto superiore. Questo implica che il terzo giunto deve essere attuato con coppia di rotazione allineata con l'avambraccio. Il quarto asse è concepito per essere coincidente con il giunto di gomito e il quinto asse coincidente con l'avambraccio, in modo da permettere la prono-supinazione del polso. L'workspace



Figura 4.3: Workspace di spalla L-Exos.

del gomito coincide con l'angolo di rotazione del quarto asse. Assunta la posizione zero per la condizione di allineamento tra l'avambraccio ed il braccio, l'ampiezza di movimento ottenuta con l' L-Exos si estende tra 2.5 e 105 gradi. Il polso dell' L-EXOS ha solo un grado di libertà non attuato, la cui ampiezza di movimento è di 180 gradi. In particolare la rotazione è possibile tra $\pm 90^{\circ}$ misurati quando il palmo della mano è allineato con il braccio e l'avambraccio nella posizione zero. Viste le considerazioni fatte, i parametri Denavit-Hartenberg per l'L-Exos sono riportati in tabella 4.1. Il principali vantaggi che l'esoscheletro L-Exos presenta e che permettono alla sua struttura di essere adatta per la riabilitazione mediante manipolatore robotico possono essere riassunte come segue:

- struttura facilmente indossabile e usabile
- è in grado di esercitare forze controllate applicate a ciascun link fino a 100N

Link (i)	d_i	a_i	α_i	q_i
0	0	0	$lpha_0$	0
1	d_1	0	$-\frac{\pi}{2}$	$q_1 - \frac{\pi}{2}$
2	0	0	$\frac{\pi}{2}$	$q_2 + \frac{\pi}{2}$
3	d_3	0	$-\frac{\pi}{2}$	q_3
4	0	0	$\frac{\pi}{2}$	q_4
5	d_5	0	0	q_5

Tabella 4.1: Parametri Denavith-Hartenberg.

- è progettato per permettere la mobilità totale all'arto superiore di un soggetto sano
- la sua progettazione meccanica è compatibile con le esigenze di pazienti con ridotte capacità di movimento, anche in termini di sicurezza e confort
- il manipolatore è capace di compensare attivamente il peso dell'arto dell'utente in quantità regolabile
- il manipolatore può essere interfacciato con applicazioni di ambiente di sviluppo in realtà virtuale per rapide prototipazione di esercizi di terapia

4.1.3 Aspetti costruttivi e meccanici

É importante premettere che un adeguato feedback aptico aiuta l'operatore di sistemi robotici a completare in maniera più affidabile compiti in realtà virtuale o di teleoperazione. I Requisiti per tali sistemi robotici son principalmente legati alla capacità di trasmettere comandi a sistemi remoti o in realtà virtuale e di restituire all'operatore stesso tutte le informazioni sensoriali quali spostamenti o forze di contatto di interazione con l'ambiente remoto o virtuale. In sostanza, la dinamica del manipolatore non deve interferire con l'interazione tra l'operatore e l'ambiente. Un'interfaccia aptica ideale si comporta come un corpo rigido, attraverso il quale l'utente interagisce con l'ambiente in tutta la gamma di frequenze delle forze in gioco nell'ambiente virtuale.

In tal senso è importante definire la trasparenza del sistema come la misura

del grado di distorsione tra le forze all'interfaccia uomo-robot e la forza di contatto desiderata comandata dall'ambiente virtuale. La trasparenza può degradare a causa di alcune caratteristiche intrinseche dei sistemi meccanici come l'inerzia, il gioco dei giunti ed i relativi attriti, la risoluzione dei sensori ed i ritardi computazionali.

Al fine di migliorare la trasparenza di utilizzo del manipolatore esoscheletrico, sono state seguite una serie di linee guida per la sua progettazione meccanica:

- posizione remota degli attuatori: questo permette di ridurre drasticamente l'inerzia percepita e perciò di aumentare la trasparenza complessiva dell'interfaccia aptica. Il posizionamento remoto degli attuatori permette inoltre di diminuire la coppia richiesta per la compensazione di gravità, permettendo di diminuire le dimensioni degli attuatori stessi e la tensione di trasmissione;
- scelta accurata dei motori con alto rapporto di coppia/peso;
- uso di trasmissione del moto tramite tendini: pur avendo bassa rigidità, possono trasmettere facilmente le coppie dei motori ai giunti distanti con zero gioco (a differenza di cinghie o ingranaggi), basso attrito e peso ridotto;
- bassi rapporti di trasmissione: questo permette di ridurre il contributo dei motori nell'inerzia percepita all'end-effector e di abbassare l'attrito di trasmissione percepito;
- realizzazione di giunti a gioco nullo;
- utilizzo di materiali leggeri per la costruzione delle parti mobili;
- garanzia di sicurezza, confort ed ergonomia nell'uso dell'interfaccia aptica

Tutti i motori dell' L-Exos son posizionati nel frame fisso (link 0). Per ciascun DoF attuato, la coppia viene trasmessa dai motori ai giunti tramite una trasmissione a tendini con cavi d'acciaio e riduttori integrati in ciascun asse del giunto. Tale accorgimento consente di ridurre le masse delle parti in movimento ed il peso della parti strutturali del manipolatore alle quali è richiesto un sostegno minore dei motori a massa ridotta che complessivamente compongono circa il 40% della massa dell'esoscheletro. Questo comporta a percepire al palmo della mano un'inerzia minore. Per le stesse ragioni son stati scelti attuatori elettrici con buoni rapporti di coppia/peso e coppia/volume. La trasmissione a tendini d'acciaio permette di garantire un basso peso ed un gioco nullo nella trasmissione della coppia, a discapito però di una bassa rigidezza meccanica. D'altro canto i sistemi di trasmissione a cavo hanno un'efficienza maggiore rispetto a quelli ad ingranaggio ed assicurano un miglior grado di "backdrivability" del si sistema.

Al fine di ottenere un'alta rigidità del manipolatore all'end-effector, dei riduttori a basso rapporto di riduzione son collocati agli assi dei giunti, permettendo di ridurre la tensione ai tendini, la loro elongazione ed il loro diametro. La riduzione stessa del diametro dei tendini portano ad un conseguente risparmio di peso e volume di tutte le parti meccaniche del sistema di trasmissione.

Grazie a questa soluzione e allo sviluppo di riduttori di velocità appositamente concepiti è stato possibile ottenere una riduzione della massa del sistema di trasmissione di circa il 35% e delle parti strutturali del 40%. I componenti strutturali (link) son stati progettati come parti a pareti sottili per poter alloggiare le parti meccaniche della trasmissione (pulegge, tendini, assali..etc) all'interno dei link, in modo tale da proteggere le parti interne da interferenze esterne, e garantendo la sicurezza all'utente del sistema esoscheletrico. Al fine di migliorare ulteriormente la leggerezza e la rigidità, i componenti strutturali son stati realizzati in fibra di carbonio, mentre altre parti in alluminio son state legate alla struttura per realizzare le connessioni con gli altri componenti meccanici.

Sistema di trasmissione

La progettazione di sistemi complessi di trasmissione del moto spesso implicano una serie di svantaggi. Mentre consentono di collocare i motori distanti dai giunti azionati, riducendo massa ed inerzia, introducono effetti quali l'accoppiamento dei giunti, attriti complessi da modellare ed elasticità dei meccanismi. Per le interfacce aptiche basate su questi sistemi di trasmissione è veramente importante realizzare un modello acculrato delle perdite d'atrito nel sistema di trasmissione stesso, in modo da poterle compensare mediante un controllo attivo.

La complessità del sistema di trasmissione risulta chiaro in figura 4.4 dove è mostrata la trasmissione del moto relativa al quarto DoF. Il sistema può essere schematizzato come una catena cinematica seriale, composta da link rigidi connessi da giunti rotoidali. Per la caratterizzazione del comportamento dinamico di tutto il sistema è conveniente separare idealmente le parti strutturali dal sistema di trasmissione composto da tendini e pulegge.

Al fine di costruire un modello del sistema di trasmissione, son state fatte le seguenti ipotesi:



Figura 4.4: Sistema di trasmissione del quarto giunto.

- la struttura del sistema può essere schematizzata come una catena cinematica seriale composta da link rigidi, collegati attraverso giunti rotoidali;
- il sistema di trasmissione è realizzato con cavi che partono dagli attuatori e passano su una serie di pulegge fino a raggiungere i giunti da attuare;
- le coppie ai giunti vengono trasmesse con deformazioni trascurabili delle parti meccaniche strutturali

Sensori di posizione e forza

La posizione dei giunti del sistema esoscheletrico L-Exos viene misurata attraverso gli encoder calettati sugli alberi di ciascun gruppo motore, mentre i segnali di forza, dovuti all'interazione tra l'utente e l'end-effector del L-Exos son stati misurati attraverso il sensore di forza triassiale sulla maniglia dell'end-effector stesso. Tutti i segnali di posizione e forza son campionati a 2 kHz.

Componenti speciali

La progettazione meccanica del sistema esoscheletrico ha incluso alcuni componenti speciali sviluppati appositamente nei laboratori del PERCRO a causa della mancanza nel mercato di componenti off-the-shelf con le performance richieste. Questo comporta il vantaggio che non solo questi componenti "custom" corrispondono alle prestazioni richieste, ma possono essere progettate per essere altamente integrate con le restanti parti meccaniche del dispositivo. Nonostante l'elevato grado di personalizzazione di questi componenti, possono essere facilmente adattati ad altri dispositivi, cosa che permette loro di essere considerati come dispositivi generici stand-alone.

I componenti sviluppati specificatamente per l'L-Exos sono i seguenti:

- 1. una guida circolare aperta;
- 2. un riduttore epicicloidale integrato
- 3. il gruppo motori

Per quanto riguarda la guida circolare, riportata in figura 4.5, è stata sviluppata specificatamente per implementare il terzo giunto rotoidale dell'L-Exos. Dal momento in cui il giunto 3 è allineato con l'asse di rotazione del braccio umano, è necessario un centro di rotazione remoto. La soluzione di adottare un cuscinetto circolare chiuso è stata sostituita con una circolare aperta, basata sulla tecnologia dei cuscineti a ricircolo di sfere . Questa soluzione innovativa permette all'utente di indossare (e svestire) facilmente l'esoscheletro, senza la necessità di inserire l'arto all'interno di una struttura chiusa, che in caso di pazienti con difficoltà motorie risulta piuttosto complicato e poco confortevole. Grazie a questa soluzione la struttura dell'L-Exos risulta aperta fino al giunto del polso, ed una volta indossata il gomito può essere attaccato alla struttura robotica attraverso delle cinghie con velcro.



Figura 4.5: La guida circolare del terzo giunto.

Oltre all'evidente vantaggio di facile indossabilità, la guida aperta permette di aumentare la mobilità del gomito durante i movimenti di abduzioneadduzione grazie all'eliminazione di parti chiuse interne che limitano l'avvicinamento del gomito stesso al tronco dell'utente, aumentando perciò il workspace dell'esoscheletro.

La guida circolare è composta principalmente da una parte fissa con geometria di una rotaia circolare ed una parte mobile, il cursore, entrambi realizzati in lega di alluminio. Il passaggio del cursore è invece realizzato con quattro piste di rotolamento in accacio AISI 404.

Per quanto riguarda invece il secondo componente speciale, il riduttore epicicloidale è l'ultimo elemento del sistema di attuazione posizionato sull'asse del giunto. L'obiettivo primario della progettazione di tale parte è la riduzione del suo peso e del suo volume rispetto ai riduttori commerciali standard. Il raggiungimento di questo obiettivo è stato pienamente ottenuto con un alto grado di integrazione con l'asse del giunto stesso.



Figura 4.6: Il riduttore integrato.

Il numero di componenti è stato ridotto imponendo al riduttore di realizzare molteplici funzioni oltre a quella d'essere unicamente un riduttore di velocità: l'albero di ingresso serve anche come puleggia condotta della trasmissione, mentre l'albero di uscita è anche l'asse di ciascun giunto. L'interposizione del riduttore tra la trasmissione a tendini e i giunti da attuare permette di ottenere anche un aumento della rigidità del sistema, attraverso la riduzione della tensione dei cavi d'acciaio che compongono la trasmissione.

La leggerezza eccezionale è stata ottenuta in parte mediante la suddetta integrazione tra le parti, ma sopratutto per mezzo di uno sforzo progettuale che ha permesso l'uso di leghe d'alluminio leggere anziché in acciaio. Gli unici componenti d'acciaio oltre ai bulloni e cuscinetti, son gli ingranaggi standard. Un grande sforzo è stato effettutato per mantenere il gioco introdotto dagli ingranaggi all'interno di un valore tollerabile. Un valore inferiore implicherebbe un aumento eccessivo dei coefficienti d'attrito del meccanismo.

Le principali caratteristiche del riduttore integrato son in seguito riportate per i due diversi modelli che son stati costruiti:

Reduction ratio:	4/6
Max torque - output:	81/30 Nm
Max Speed - input:	1430 <i>rpm</i>
Max Load Time:	120 s
Mass:	342 - 347 g , depending on the axis lenght
D:	73 mm
L - without axis:	42 mm
Backlash - output:	< 8 arcmin
Efficiency:	0.93

I primi quattro gradi di libertà del manipolatore vengono attuati da quattri identici gruppi motore. Ciascun gruppo è dotato di un Motore DC a magneti permanenti (VERNITRON 3730) senza telaio e di un encoder ottico ad alta risoluzione (Heidenhain rod1020, 1024 tacche). In ciascun gruppo motore è inclusa la puleggia di trasmissione del moto ai tendini direttamente vincolata al rotore del motore stesso. La puleggia motrice è supportata da due cuscinetti a sfere situati in corrispondenza delle due pareti del link fisso (link 0). Tutte le parti strutturali del gruppo motore son costruite con leghe leggere (Ergal 7075).



Figura 4.7: Schema del gruppo motore.

Le principali caratteristiche del gruppo motore sono in seguito riportate:

Coppia continua:	2 Nm
Coppia massima:	3.7 Nm
Peso:	1150 g
D:	110 mm
L:	140 mm

Prestazioni del sistema L-Exos

Le prestazioni raggiunte dal sistema esoscheletrico L-Exos son riportate in tabella 4.1.3

Carico utile (forza continua):	50 N
Carico utile (forza massima):	100 N
Backlash:	10mm all'end-effector
Peso:	11Kg
Rigidità (stimata):	3 N/mm
Rigidità (misurata):	3 N/mm
Workspace:	70% dell'arto superiore umano

Il peso complessivo dell'esoscheletro L-Exos è di circa 11Kg, di cui 6Kg sono distribuiti sul link 0. Il peso complessivo è principalmente dovuto alla massa dei 4 gruppi motori. Il rapporto di peso/carico utile risulta essere piuttosto basso (100 N contro 11Kg di massa). Il valore di rigidezza stimata rappresenta la condizione peggiore teorica.

4.2 Segnali elettromiografici

I segnali sEMG sono stati acquisiti con misura differenziale tra due elettrodi sulla superficie della pelle, al centro del ventre muscolare interessato. Per ottenere un'alta qualità del contatto tra elettrodo e pelle, son state seguite le raccomandazioni del SENIAM [43]. In particolare all'inizio della sessione di misura, la pelle del soggetto è stata preparata mediante una pasta abrasiva Nuprep (Weaver and Company) per diminuire l'impedenza di contatto tra cute ed elettrodo. Gli elettrodi utilizzati per prelavare il segnale sono di tipo Ag/AgCl pregellati in Foam con diametro conduttivo di 16 mm. La distanza tra gli elettrodi è stata fissata a 20 mm.

I segnali sEMG acquisiti riguardano i muscoli di spalla e gomito. La posizione degli elettrodi è mostrata in figura 4.8. Per il giunto di spalla sono stati acquisiti i segnali elettromiografici di superficie dei seguenti gruppi muscolari:

• Pettorale maggiore (PM);

- Deltoide anteriore (DA);
- Deltoide posteriore (DP).

Per il giunto del gomito son stati acquisiti i segnali elettromiografici di superficie dei seguenti gruppi muscolari:

- Bicipite Brachiale (BB);
- Tricipite Brachiale (TB).



Figura 4.8: Posizione degli elettrodi sEMG su braccio e spalla. Vista anteriore (a.) e posteriore (b.) dei braccio e spalla.

E stato scelto di usare due elettrodi di riferimento separati per i due giunti. Per i gruppi muscolari della spalla ed il segnale ECG, l'elettrodo di riferimento è stato posizionato al centro dell'osso della scapola destra, punto privo di massa muscolare, mentre per quelli del gomito è stato posizionato all'altezza dell'epicondilo, un punto privo di massa muscolare. Per evitare il più possibile gli artefatti dovuti al movimento degli elettrodi e dei cavi, questi ultimi son stati fissati alla pelle per mezzo di fasce o nastro adesivo.

Il segnale sEMG infine è stato campionato con frequenza pari a $f_c = 256$ Hz con risoluzione di 24bit, ed infine filtrato con filtro digitale passabanda 2-100 Hz. La banda utile del segnale sEMG è stata scelta in funzione delle seguenti motivazioni:

• la frequenza passa-alto è stata scelta dell'ordine di 2 Hz per eliminare gli artefatti dovuti al movimento degli elettrodi;

• la frequenza passa-basso è stata scelta a 100 Hz perchè l'informazione di forza contenuta nel segnale sEMG risiede in una banda di frequenze che non supera i 150 Hz, ma in particolare nello studio condotto, i segnali elettromiografici riportano contenuti significativi fino a frequenze di 100 Hz, a causa delle contrazioni isometriche in condizioni quasi statiche, con forze relativamente basse rispetto alla massima contrazione volontaria (MVC) isometrica di ciascun gruppo muscolare.

L'ambiente di misura intensamente rumoroso ha portato alla necessità dell'aggiunta di un filtro Notch con frequenza centrata a 50 Hz per eliminare il rumore dovuto all'alimentazione elettrica. Il segnale sEMG del pettorale maggiore (PM) risente inoltre del disturbo di crosstalk di ECG, ed è dunque necessario filtrarlo tramite la procedura descritta nel paragrafo 4.2.2.

4.2.1 Amplificatore g.USBamp

Tutti gli elettrodi son stati connessi all'amplificatore g.USBamp (GTec), amplificatore multimodale di biosegnali, per qualsiasi tipo tipo di segnali elettrofisiologici quali EEG, elettrocardiogramma (ECG), EOG, elettromiografia (EMG) e altri sensori esterni. L'apparecchiatura ha un convertitore ADC digitale/analogico a 24-bit, ed un sistema di trasmissione dati che può essere connesso direttamente al PC tramite USB. Il dispositivo è caratterizzato da un eccellente qualità di acquisizione del segnale, e bassi livelli di rumore. È possibile utilizzare uno stack di più dispositivi (di 16 canali ciascuno) per ottenere un sistema multicanale. Il dispositivo ha la certificazione di apparecchiatura medica, ed ha le certificazioni CE e FDA. Gli ingressi analogici dell'hardware sono suddivisi in quattro blocchi, ciascuno con quattro ingressi di segnale, uno di riferimento e uno di terra. Oltre questi, la scheda di acquisizione permette di avere un canale triggerato che viene campionato in contemporanea con i canali analogici, 2 ingressi digitali e 2 uscite digitali. La casa costruttrice oltre al all'hardware fornisce un pacchetto di software che comprende i driver a 32/64 bit, software di analisi dei segnali elettromiografici, ECG, EEG e tutti i toolbox per Matlab.

Highlights:

- Certificazione CE e FDA come dispositivo medico per l'uso umano, secondo le normative mediche EN 60601-1 (IEC 60601-1);
- Registrazione di segnali EEG, ECoG, ECG, EMG e EOG e trasmissione tramite USB;
- 16 ingressi analogici campionati con un convertitore a 24 Bit e frequenze di campionamento fino a 38400 Hz per canale;



Figura 4.9: Sistema di acquisizione USBamp (g.tec medical engineering).

- Filtraggio digitale dei dati dei biosegnali (DC 2.4 kHz);
- Over-sampling per l'aumento del rapporto segnale rumore;
- Sample & Hold simultaneo per tutti i canali;
- Connessione diretta degli elettrodi con connettori di sicurezza standard o sistemi di connessione per una veloce applicazione degli elettrodi;
- Possibilità di utilizzo per la registrazione diretta dei segnali encefalografici (ECoG) e cardiaci (CF-system);
- Analisi on-line dei segnali su Simulink;
- Possibilità di essere combinato con g.BSanalyze (toolboox) per l'analisi off-line dei biosegnali su MATLAB.

4.2.2 Eliminazione disturbo ECG

L'esempio più evidente di crosstalk muscolare è quello della sovrapposizione del tracciato ECG del cuore al segnale elettromiografico dei muscoli del tronco e della spalla, in particolare sul lato sinistro del corpo umano. La particolarità di questo genere di disturbi rende la loro reiezione più complicata rispetto ad altri. Come si può notare in figura 4.10 il cuore ha uno spettro con potenza concentrata entro i 100 Hz, con la maggior concentrazione centrata tra i primi 35 Hz. Il Segnale elettromiografico ha la maggior
parte della potenza concentrata tra i 30 e 150 Hz, e questo porta ad una sovrapposizione piuttosto marcata tra i due spettri e dunque la evidente difficoltà di filtraggio con i metodi classici. La forza della contrazione cardiaca, e dunque il relativo segnale elettromiografico, raggiunge nell'area di misura un'ampiezza elettrica molto marcata nei picchi, che raggiungono i livelli di contrazione massima di alcuni dei gruppi muscolari più forti corpo umano, fino ai 1500-2000 μV . Un'altra caratteristica tipica del cuore, è la contrazione ripetuta che in condizioni normali oscilla tra i 60 e 100 BPM, ma che non risulta esattamente costante anche tra periodi brevi, nonostante conservi inalterate tutte le caratteristiche di forma del complesso QRS e dell'onda precedente P e seguente T. Questo porta alla facile individuazione visiva del disturbo all'interno del tracciato elettromiografico, e rende possibile l'utilizzo di tecniche di riconoscimento del disturbo a template di forma.



Figura 4.10: Spettro durante il periodo di EMG a riposo ("ECG") e durante una contrazione.

Molti lavori hanno puntato a massimizzare l'eliminazione di questo disturbo, con l'intento di lasciare inalterata l'informazione elettromiografica [44], con metodi spesso molto differenti tra loro, come nel caso del [45] dove viene usata una tecnica di Adaptive ECG spike clipping migliorata con l'utilizzo di filtri passa alto ottimi, oppure l'utilizzo della tecnica ICA (Independent component analysis) [46][47], in altri casi metodi che portano all'utilizzo di filtri adattativi [48][49] o altri che usano la tecnica di sottrazione tramite il riconoscimento di template di forma del disturbo ECG.

I punti critici di alcuni di questi metodi sono essenzialmente quattro. In primo luogo, il disturbo che la contrazione muscolare induce sul segnale EMG superficiale si trasmette in maniera particolarmente non lineare, a causa dell'anisotropia dei tessuti biologici del corpo umano frapposti tra cuore e gruppo muscolare interessato (ossa, altri gruppi muscolari, sistema linfatico..etc). Il tracciato ECG interferisce inoltre con ampiezze a volte superiori al gruppo muscolare interessato. In secondo luogo nelle applicazioni di controllo di sistemi robotici riabilitativi è richiesta l'eliminazione Real-Time del disturbo, cosa che si rivela impossibile per alcuni degli algoritmi sopra elencati. Terzo punto critico è la necessità in molti casi della misura (e retroazione) del segnale ECG, con l'aggiunta quindi di un ulteriore paio di elettrodi e dunque di minor vestibilità del sistema e di confort da parte del paziente. Ultimo punto, è l'effetto di shift frequenziale (verso le basse frequenze) che subisce il segnale elettromiografico con l'aumento progressivo del numero di ripetizioni di un esercizio fisico o riabilitativo, e dunque una risultante maggior sovrapposizione tra lo spettro EMG e quello ECG. Il cuore non risente del fenomeno dell'affaticamento e dunque il suo spettro risulta pressochè invariante rispetto alla fatica muscolare.

In questo lavoro di tesi viene proposto un metodo real time per la sottrazione del rumore ECG dal segnale elettromiografico. Il metodo richiede la misura del segnale cardiaco e dunque l'aggiunta di due elettrodi per la sua misura differenziale. In primo luogo, vengono misurati sia i segnali elettromiografici dei gruppi muscolari interessati e sia il tracciato ECG, in una situazione di confort e riposo per il paziente. L'idea è quella di portare il segnale ECG nel sistema di riferimento del disturbo indotto nel gruppo muscolare interessato, in modo tale da poterlo cancellare in controfase. Per far ciò bisogna equalizzare il segnale ECG ed eliminare il ritardo di trasmissione del disturbo che si propaga dalla zona sternale-costale sinistra fino al gruppo muscolare, ed infine sottrarlo dal segnale EMG corrotto dal disturbo.

L'inconveniente di dover utilizzare il segnale ECG sta nel fatto che anche questo segnale è a sua volta disturbato dal crosstalk muscolare del pettorale sinistro che durante l'esercizio fisico potrebbe non essere sempre perfettamente a riposo. Per eliminare il più possibile il disturbo dovuto alle sue contrazioni muscolari, il segnale ECG viene prima filtrato passa-basso, con un filtro butterworth digitale del secondo ordine, con frequenza di taglio pari a 25 Hz.

Il metodo proposto dunque si avvale della più semplice tecnica di identificazione, la stima della risposta impulsiva discreta tramite minimi quadrati. Presi un numero di campioni sufficienti del segnale EMG y_k corrotto dal disturbo ECG e lo stesso numero di campioni per quanto riguarda il segnale ECG puro u_k , e supposto che per la descrizione della dinamica di trasmissione sia sufficiente un modello di predizione di tipo FIR di ordine p, possiamo scrivere:

$$y_k = \sum_{i=0}^t h_{k-i} u_i = \sum_{j=0}^\infty h_j u_{k-j}$$
(4.1)

$$y_k = \sum_{j=0}^{\infty} h_j u_{k-j}; \quad \text{con} \quad u_k = 0, \forall k < 0$$
 (4.2)

$$\begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_{k-1} \\ y_k \end{bmatrix} = \begin{bmatrix} u_0 & 0 & \cdots & 0 \\ u_1 & u_0 & 0 & \vdots \\ \vdots & \vdots & \ddots & \ddots \\ & & & u_0 & 0 \\ & & & & & u_0 \\ & & & & & & u_0 \\ & & & & & & \vdots \\ u_k & u_{k-1} & \cdots & & u_{k-p} \end{bmatrix} \begin{bmatrix} h_0 \\ h_1 \\ \vdots \\ h_{p-1} \\ h_p \end{bmatrix}$$
(4.3)

Per un istante k, il predittore assume quindi la semplice forma:

$$\hat{y} = H(k)\hat{\theta} \tag{4.4}$$

 $\operatorname{con} H(k)$ pari a:

$$H(k) = [u(k-d)\cdots u(k-d-p+1)]$$
(4.5)

Il vettore riga H(k) è chiamato anche regressore e $\hat{\theta}$ sono i parametri da trovare, che non sono altro che la risposta impulsiva h(k) del sistema di trasmissione del disturbo cardiaco.

L'errore di predizione che si otterrà con questo modello è descrivibile nella forma:

$$\varepsilon(k) = y(k) - \hat{y}(k) = y(k) - H(k)\overline{\theta}$$
(4.6)

La relazione 4.4 non è, in generale direttamente, utilizzabile per stimare $\hat{\theta}$, poiche è necessario avere un numero di misure sufficiente per stimare il vettore dei parametri della risposta impulsiva. Per ottenere ciò è necessario avere un numero sufficientemente alto di istanti di misura: $y(k-l), y(k-l+1), \dots, y(k)$. definito il vettore $\mathbf{y}(k)$ come segue:

$$\underline{\mathbf{y}}(k) = \begin{bmatrix} y(k-l) \\ y(k-l+1) \\ \vdots \\ y(k) \end{bmatrix}$$
(4.7)

In maniera analoga si può definire la matrice $\mathbf{H}(k)$ impilando il regressore per ciascun istante di tempo di misura:

$$\mathbf{H}(k) = \begin{bmatrix} H(k-l) \\ H(k-l+1) \\ \vdots \\ H(k) \end{bmatrix}$$
(4.8)

Il vettore impilato di errori di predizione si può scrivere dunque nella forma:

$$\underline{\varepsilon}(k) = \underline{\mathbf{y}}(k) - \underline{\mathbf{\hat{y}}}(k) = \underline{\mathbf{y}}(k) - \mathbf{H}(\mathbf{k})\hat{\theta}$$
(4.9)

L'indice scalare rispetto al quale si calcola la soluzione ottima è la somma al quadrato degli errori di predizione:

$$J = \underline{\varepsilon}(k)^T \underline{\varepsilon}(k) = [\underline{\mathbf{y}}(k) - \mathbf{H}(\mathbf{k})\hat{\theta}]^T [\underline{\mathbf{y}}(k) - \mathbf{H}(\mathbf{k})\hat{\theta}]$$
(4.10)

Il minimo dell'indice si trova facendone la derivata rispetto ai parametri $\hat{\theta}$ e studiando quando si annulla:

$$\frac{\partial J}{\partial \hat{\theta}} = -2\underline{\mathbf{y}}(k)^T \mathbf{H}(\mathbf{k}) + 2\hat{\theta}\mathbf{H}(\mathbf{k})^T \mathbf{H}(\mathbf{k}) = 0$$
(4.11)

Supponendo invertibili le matrici interessate si ottiene la soluzione ai minimi quadrati e dunque si può concludere l'identificazione:

$$\hat{\theta} = \left(\mathbf{H}(\mathbf{k})^T \mathbf{H}(\mathbf{k})\right)^{-1} \mathbf{H}(\mathbf{k})^T \underline{\mathbf{y}}(k).$$
(4.12)

Inoltre, per un modello di questo tipo possiamo fare uno studio di crosscorrelazione ingresso-uscita, descrivibile come segue:

$$y_{k+i} = \sum_{j=0}^{\infty} h_j u_{k+i-j}$$
(4.13)

$$r_{uy}(i) = \lim_{N \to \infty} \frac{1}{2N+1} \sum_{k=-N}^{N} u_k \sum_{j=0}^{\infty} h_j u_{k+i-j}$$
$$= \sum_{j=0}^{\infty} h_j \left(\lim_{N \to \infty} \frac{1}{2N+1} \sum_{k=-N}^{N} u_k u_{k+i-j} \right) = \sum_{j=0}^{\infty} h_j r_{uu}(i-j) \quad (4.14)$$

dove r_{uu} è l'autocorrelazione dell'ingresso.

La cross-correlazione In/out dipende perciò dalla convoluzione tra la risposta impulsiva e l'autocorrelazione dell'ingresso.



Figura 4.11: Modello per la stima della risposta impulsiva h(t) tra EMG a riposo corrotto dal disturbo ECG e lo stesso.

Se supponiamo di essere in presenza di un rumore di misura e/o l'uscita dovuta ad altre frazioni di ingresso scorrelate dal resto e a media nulla, possiamo scrivere come segue:

$$y(k) = y'(k) + e(t)$$
(4.15)

dove e(t) è un rumore di misura e/o l'uscita dovuta ad altre frazioni di ingresso scorrelate dal resto ed a media nulla. In questo caso la cross-correlazione In/Out possiamo scriverla come segue:

$$r_{uy}(i) \approx \frac{1}{2N+1} \sum_{k=-N}^{N} u_i(y'_{k+i} + e_{k+i}) = r'_{uy}(k) + r_{ue}(k) = r'_{uy}$$
(4.16)

Il fatto che il rumore di misura sia scorrelato dall'ingresso comporta che la cross-correlazione relativa sia nulla.

Una volta identificato il modello di trasporto del disturbo ECG sul segnale EMG del gruppo muscolare interessato, il segnale ECG puro passa dunque attraverso il filtro FIR i cui parametri sono esattamente quelli trovati con l'identificazione, ed infine viene sottratto dal segnale EMG per eliminarne l'effetto. Per migliorare la reiezione del disturbo il segnale risultante viene ancora una volta filtrato con un filtro butterworth passa-alto del quarto ordine con frequenza di taglio pari a 20 Hz. In figura 4.12 è mostrato il diagramma a blocchi del sistema di eliminazione del disturbo di crosstalk dell'ECG.

In figura 4.13 viene mostrato un tratto di segnale elettromiografico corrotto dal disturbo indotto dell'ECG. Di seguito in figura 4.15 vengono mostrati i risultati ottenibili con questa metodologia ed il suo confronto rispetto alla tecnica convenzionale di reiezione del disturbo ECG, ovvero i filtri passa-alto.



Figura 4.12: Diagramma a blocchi del sistema di eliminazione del disturbo di crosstalk ecg dai segnali sEMG.

4.3 Schema di acquisizione

Lo schema di acquisizione è stato concepito per gestire l'acquisizione tutti i segnali utili allo studio da svolgere, ed in particolare: i segnali elettromiografici dei gruppi muscolari interessati tramite il sistema di acquisizione GTEC Usb.Amp; i segnali di posizione dei 5 giunti dell'LExos tramite gli encoder sul sistema di trasmissione, quelli di forza all'end-effector misurati tramite il sensore di forza triassiale posizionato sulla maniglia dell'end-effector. Tutti i segnali relativi alla cinematica del LExos ed alle forze vengono acquisiti dal computer industriale che controlla l'esoscheletro stesso, ed infine inviati tramite protocollo UDP al pc utilizzato per l'acquisizione e la sincronizzazione di tutti i dati.

Lo schema di acquisizione sviluppato secondo questi requisiti è composto di tre blocchi funzionali, ciascuno incaricato di un compito specifico:

- un sistema di selezione e gestione automatica della raggiungibilità dei punti da raggiungere che permette la selezione manuale o automatica dei punti previsti dal protocollo di misura e gestisce la loro raggiungibilità. Una volta raggiunto il punto d'interesse, l'algoritmo si occupa di far iniziare la registrazione del dataset.
- un sistema di gestione delle fasi di acquisizione e del protocollo di misura scelto, il cui compito è quello di automatizzare tutte le fasi del protocollo ed il conteggio del numero di ripetizioni eseguite per il movimento previsto dal protocollo.
- una interfaccia grafica che permette di guidare il soggetto durante la fase di acquisizione, visualizzando sullo schermo le informazioni sulla fase di movimento del protocollo da svolgere, il numero di ripetizioni



Figura 4.13: Sopra il segnale sEMG del pettorale maggiore destro, corrotto dal crosstalk ECG. Sotto il tracciato ECG.

eseguite del movimento previsto dalla fase del protocollo ed il punto raggiunto dall'end-effector attorno al quale compiere i movimenti.

Lo schema di acquisizione sul centro di calcolo permette di sincronizzare i dati sEMG e i segnali provenienti dal L-Exos ad una frequenza di 256 Hz.



Figura 4.14: Segnale EMG corrotto dal crosstalk ECG e Segnale filtrato con la cancellazione del disturbo tramite identificazione FIR.



Figura 4.15: Confronto tra il MAV del segnale EMG grezzo, quello filtrato con identificazione FIR e quello con la tecnica standard passa-alto a 35 Hz.

Capitolo 5

Predizione di coppia con elettromiografia

5.1 Perchè usare le reti neurali

Il corpo umano è un tipico sistema fuzzy complesso, perciò la misura dei suoi segnali biologici come ad esempio quello elettromiografico di superficie contiene informazioni e logiche particolarmente sfumate (fuzzy appunto). E molto difficile ottenere lo stesso segnale EMG per lo stesso movimento anche con lo stesso soggetto. Inoltre ogni attività muscolare per un certo movimento è altamente non lineare, in quanto la responsabilità di ogni muscolo per il movimento considerato varia in base agli angoli articolari. Un muscolo non si occupa solo di un tipo di movimento, in base alla postura si impegna a stabilizzare l'articolazione stessa o ad aiutare altri movimenti. Oltre questo, il livello di attività di ciascun muscolo ed il modo di utilizzarlo per un certo movimento è differente tra soggetti diversi. Il livello di attività del muscolo è inoltre influenzato dalle condizioni fisiologiche del soggetto. In aggiunta il livello di attività di alcuni muscoli (ad esempio i biarticolari) è influenzato dal movimento delle altre articolazioni contigue in base a come si distribuisce il carico di lavoro. Lo stesso carico di lavoro esterno complessivo influenza l'attività muscolare in maniera non lineare. Pertanto, per controllare dei manipolatori tramite segnali EMG di superficie è necessario usare un controllo non lineare, flessibile e con buona capacità di adattamento all'utente che lo utilizza. In aggiunta bisogna considerare l'interazione tra il manipolatore e l'utente stesso, concentrando l'attenzione sul fatto che il robot deve avere capacità di controllo in tempo reale, per evitare che eventuali ritardi di assistenza robotica facciano gravare troppo carico di lavoro sull'utente stesso, o che l'interazione tra robot e utente risulti complessivamente instabile a causa

di questi, o ancor più semplicemente che l'utilizzo risulti poco confortevole a causa di movimenti poco naturali.

5.2 Funzionamento delle reti neurali

La rete neurale è un sitema dinamico avente la topologia di un grafo orientato con nodi, i neuroni artificiali, ed archi, i pesi sinaptici. Il termine rete è riferito alla topologia dei collegamenti tra i neuroni. Le reti neurali artificiali sono nate essenzialmente per riprodurre attività tipiche del cervello umano, come il riconoscimento di forme, il riconoscimento vocale, e tutte quelle esigenze di classificazione, di approssimazione o predizione tipiche del cervello umano. L'idea è quella di imitare il comportamento delle cellule neurali e le connessioni sinaptiche, creando un calcolatore che simile al cervello sia non-lineare e parallelo, costituito da elementi di elaborazione molto semplici (i neuroni) e da connessioni e pesi. Il cervello è in grado di modificare le connessioni tra i neuroni in base all'esperienza acquisita, rendendolo in grado di imparare. La particolare struttura del cervello ha la caratteristica di non avere un controllo centralizzato, bensì le zone del cervello funzionano insieme, influenzandosi reciprocamente e contribuendo alla realizzazione di uno scopo specifico. Per questo motivo, ed anche grazie alla plasticità del cervello, può ovviare alla mancanza di connessioni, senza degradare significativamente le prestazioni.

Per riprodurre artificialmente il cervello perciò occorre realizzare una rete di elementi semplici, con una struttura distribuita, pesantemente parallela, capace di apprendere e infine di generalizzare ciò che ha appreso, ovvero di riprodurre uscite sensate in corrispondenza di ingressi non incontrati durante l'addestramento e quindi complessivamente eseguire delle azioni senza la necessità di "pensare", basate solo sulle associazioni imparate.

5.2.1 Modello matematico del Neurone

Il Neurone artificiale (vedi figura 5.1) ha n ingressi ed una sola uscita. Ogni ingresso ha associato un peso w_i che determina la conducibilità del canale di ingresso. L'attivazione a del neurone è una funzione della somma pesata degli ingressi. L'uscita y è la trasmissione dell'attività neurale all'esterno, è calcolata applicando la funzione di attivazione alla somma pesata degli ingressi. Nel modello del neurone è inclusa anche una soglia θ , che ha l'effetto di abbassare il valore in ingresso alla funzione di attivazione. La funzione di attivazione f è detta anche funzione di trasferimento.



Figura 5.1: Modello matematico di un neurone artificiale.

Complessivamente possiamo scrivere:

$$a = \sum_{i=1}^{n} w_i x_i \tag{5.1}$$

$$y = f(a) = f\left(\sum_{i=1}^{n} w_i x_i - \theta\right)$$
(5.2)

Interpretando la soglia come il peso associato ad un ulteriore canale di ingresso x_0 , di valore sempre costante pari a -1, possiamo scrivere:

$$y = f(a) = f\left(\sum_{i=0}^{n} w_i x_i\right)$$
(5.3)

$$\operatorname{con} w_0 = \theta \tag{5.4}$$

La funzione di trasferimento di un neurone può essere di diversi tipi e la sua scelta accurata è molto importante in base al tipo di applicazione nella quale si vogliono usare le reti neurali. Le funzioni a soglia più comuni vengono descritte come segue:

• Funzione a soglia:

L'uscita del neurone usa una funzione di attivazione a soglia molto semplice che restituisce un valore unitario se l'attivazione supera la soglia θ , o nullo altrimenti:

$$y = \begin{cases} 1 & \text{se} \quad \sum_{i=1}^{n} w_i x_i \ge \theta \\ 0 & \text{se} \quad a < \theta \end{cases}$$
(5.5)



• Funzione lineare:



• Funzione Sigmoide logaritmica: L'uscita del neurone è calcolata come segue:

$$y = \text{logsig}(n) = \frac{1}{1 + e^{-n}}$$
 (5.6)



• Funzione Tangente Iperbolica (TanSig): L'uscita del neurone è calcolata come segue:

$$y = \tanh(n) = \frac{1 - e^{-2n}}{1 + e^{-2n}}$$
(5.7)

5.2.2 Architettura di una rete neurale

Per definire una rete neurale, oltre al modello per ciascun neurone, è estremamente importante definire la giusta architettura della rete. Le architetture delle reti neurali sono essenzialmente di due tipi, reti completamente connesse (non stratificate) e quelle stratificate. Nel primo caso, ogni neurone è



connesso in modo bidirezionale con tutti gli altri. Nel caso delle reti stratificate si possono individuare degli strati di neuroni tali che ogni neurone è connesso con tutti quelli dello strato successivo, ma non esistono connessioni tra i neuroni all'interno dello stesso strato, né tra neuroni di strati non adiacenti.



Figura 5.2: a) Architettura di rete completamente connessa, b) Architettura di rete stratificata.

Nel caso delle reti stratificate, il numero degli strati e dei neuroni per strato dipendono fortemente dallo specifico problema che si intende risolvere. Non esistono regole generali per decidere il numero di neuroni, ed in generale nemmeno il numero degli strati, ma in base al problema che si vuole risolvere ci sono delle linee guida che dipendono dal numero degli esempi di ingresso/uscita che si vogliono dare alla rete per la fase di apprendimento.

Nello strato di ingresso non avviene alcuna computazione, ha il semplice compito di passare gli ingressi allo strato successivo. Gli strati intermedi tra quello di ingresso e quello di uscita son chiamati strati nascosti (o hidden). Inoltre dato che i segnali viaggiano dallo strato di ingresso verso lo strato di uscita, si parla di rete feedforward.

La caratteristica principale degli strati nascosti è il fatto che non comunicano direttamente con l'esterno ma permettono alla rete di costruire delle opportune rappresentazioni interne degli stimoli in ingresso in modo da facilitare il compito della rete. Le connessioni tra i neuroni di una rete stratificata sono rappresentate mediante un numero di matrici pari al numero di coppie di strati adiacenti, ciascuna delle quali contiene i pesi delle connessioni tra le copie di neuroni di due strati adiacenti.

5.2.3 Reti Feedforward

Le reti feedforward come già detto sono delle reti neurali stratificate, nelle quali si possono indentificare uno strato di ingresso, degli strati nascosti ed uno strato di uscita, ciascuno con un certo numero di neuroni. Le reti feedforward possono essere di vario tipo, in base al tipo di applicazione per la quale vengono utilizzate e soprattutto per il tipo di funzioni di trasferimento che vengono utilizzate nei neuroni. Ci sono reti Multilayer Perceptron (MLP) che hanno la caratteristica di avere strati di neuroni con funzione di trasferimento a gradino con soglia, utili principalmente nel campo della classificazione, in quanto capaci di dividere lo spazio delle features in ingresso identificando dei semipiani, dei poligoni o delle regioni in grado di raggrupparle e dividerle in base alla classe. In figura 5.4 si può intuire il loro funzionamento.



Figura 5.3: Esempio di rete neurale Feedforward con uno strato d'ingresso, due hidden ed uno di uscita.

Oltre alle reti MLP son comuni per scopi di classificazione le reti a basi radiali (RBF), dove la funzione di trasferimento è tipicamente di tipo gaussiano o simile. La struttura è a tutti gli effetti una rete feedforward con strati hidden che realizzano funzioni base $\Phi_i(x)$ circolari, centrate sui punti dello



Figura 5.4: a) Singolo perceptron, capace di dividere linearmente il piano e le classi in due semipiani; b) MLP a 3 layer (ingresso-hidden-uscita) e due neuroni hidden, capace di delimitare una banda o un settore di piano tra due rette; c) MLP 3 layer, 3 neuroni hidden, capace di delimitare regioni quasi poligonali; d) MLP 3 layer 5 neuroni hidden, capaci di creare delimitare la classe in un poligono a 5 lati; e) MLP con due layer hidden, capaci di associare alla stessa classe due poligoni.

spazio di input, in modo da raggruppare insieme solo i punti simili tramite cluster.

Le reti feedforward in generale hanno l'utilità di poter essere usate per comprendere una funzione utilizzando solo le osservazioni sui dati, cosa particolarmente utile nelle applicazioni in cui la complessità della funzione e la difficoltà dell'elaborazione rende la progettazione di una tale funzione impraticabile con i normali procedimenti di analisi manuale. Questa caratterista rende l'utilizzo delle reti feedforward particolarmente vantaggioso in applicazioni di approssimazione delle funzioni o di regressione, tra cui la previsione di serie temporali e la modellazione dei sistemi.

Time Delay Neural Network

Le reti RBF oltre all'utilità nelle applicazioni di classificazione, son piuttosto utili anche nell'interpolazione di funzione e nella stime di serie temporali, anche se per input molto distanti dal dataset di training hanno prestazioni che, a causa della classica forma a campana delle gaussiane, tendono a degradare rapidamente.

Specifiche per questo genere di applicazioni son le reti feedforward con funzioni base sigmoidi, come ad esempio le sigmoidi a tangente iperbolica, capaci di approssimare ed imparare le relazioni non-lineare che legano i vettori di ingresso con quelli di uscita. Tipicamente una rete che ha il compito di approssimare delle funzioni è costituita da uno strato d'ingresso, degli strati hidden con funzione di trasferimento *Tan-Sig* ed uno strato di uscita con neuroni con funzione di trasferimento lineare. Una rete così fatta è un approssimatore generale di funzioni, capace di approssimare arbitrariamente bene funzioni con un numero finito di discontinuità se il numero di neuroni negli strati hidden è sufficiente.

Nel caso si voglia approssimare o fare predizione di serie temporali è necessario utilizzare delle reti dinamiche, con le uscite dipendenti non solo dagli ingressi correnti (come nel caso delle reti feedforward classiche), ma anche dagli istanti precedenti di ingressi, uscite e gli stati della rete stessa. Una rete di questo tipo è molto simile alle classiche reti feedforward a parte per il fatto che lo strato di ingresso (ed eventualmente quelli hidden) hanno come ingresso di attivazione dei neuroni anche un certo numero di istanti precedenti oltre a quelli correnti. Questo risulta utile se si vuole modellare un sistema dinamico e si voglia tenere conto della dinamica interna degli stati del sistema da modellare.

Reti neurali dinamiche di questo tipo sono anche chiamate *Time Delay Neural Network*, e si suddividono in due categorie: quelle che hanno solo connessioni feedforward e quelle che che hanno connessioni ricorrenti, feedback. La rete dinamica più semplice è chiamata Focused Time-Delay Neural Network (vedi figura 5.5), che ha i ritardi all'ingresso del primo strato hidden.

5.2.4 Addestramento delle reti Feedforward

Apprendere significa migliorare la capacità di esecusione di un certo compito attraverso l'esperienza. Perchè la rete possa apprendere ha dunque bisogno di entrare a contatto con il problema o eventualmente ad una serie di esempi dai quali estrarre le informazioni utili e trasformarle in esperienza. Prima dell'apprendimento la rete neurale non ha nessuna forma di conoscenza, perciò bisogna insegnarle come comportarsi con l'input che riceve. Grazie all'ad-



Figura 5.5: Focused Time-Delay Neural Network con uno strato di ingresso, uno hidden ed uno di uscita.

destramento della rete, la conoscenza viene memorizzata sui pesi della rete, tramite algoritmi di ottimizzazione che ne fanno variare i valori cercando di minimizzare una certa funzione di errore.

In generale ci sono tre grandi paradigmi di apprendimento, ciascuno corrispondente ad un particolare compito di apprendimento, quello supervisionato, quello non supervisionato e l'apprendimento per rinforzo.

L'apprendimento supervisionato prevede che alla rete venga presentato un training set preparato da un supervisore esterno, composto da un numero sufficiente di coppie di ingressi-uscite attese. La risposta fornita dalla rete per ogni esempio viene confrontata con la risposta desiderata, si valuta la differenza (errore) fra le due e, in base a tale differenza, si aggiustano i pesi. Questo processo viene ripetuto sull'intero training set finchè le uscite della rete producono un errore al di sotto di una soglia prestabilita.

L'apprendimento non supervisionato è basato su algoritmi che modificano i pesi della rete facendo riferimento esclusivamente ad un insieme di dati per i soli ingressi. Questi algoritmi hanno la capacità di raggruppare i dati di ingresso e di individuare degli opportuni cluster rappresentativi dei dati stessi, facendo uso di metodi topologici o probabilistici. Algoritmi di questo tipo vengono usati tipicamente per tecniche di compressione dei dati.

A differenza dei primi due tipi di addestramento, quello per rinforzo si prefigge lo scopo di individuare un certo comportamento a partire da un processo d'osservazione dell'ambiente esterno. Ogni azione ha un impatto sull'ambiente che a sua volta produce una retroazione che guida l'algoritmo stesso nel processo di apprendimento. Questa classe di problemi postula la presenza di un agente, dotato di capacità di percezione, che esplora un ambiente e nel quale intraprende una serie di azioni. L'ambiente a sua volta fornisce in risposta un incentivo o un disincentivo secondo i casi. L'apprendimento per rinforzo tenta dunque di massimizzare gli incentivi cumulati ricevuti dall'agente nel corso della sua esplorazione del problema. A differenza dell'apprendimento supervisionato, non sono mai presentate delle coppie di ingresso ed uscita di esempi noti e ne si procede alla correzione esplicita di azioni sub-ottimali. L'algoritmo perciò è focalizzato sulle prestazioni online, cercando di bilanciare l'esplorazione di situazioni ignote e lo sfruttamento della conoscenza corrente. A tutti gli effetti questo tipo di addestramento è estremamente simile all'insegnamento che i genitori danno ai neonati, incentivando o "punendo" le azioni che il neonato fa durante l'esplorazione del mondo e dei comportamenti.

Per quanto riguarda l'addestramento di tipo supervisionato, è fondamentale che il set di esempi ingresso-uscita possano coprire a sufficienza la variabilità del problema, soprattutto nel caso in cui si abbia a che fare con sistemi particolarmente non lineari. Si può affermare quindi che la rete ha più probabilità di comprendere facilmente il problema se il set di dati di apprendimento risulta sufficiente sia in numero di esempi che in quantità di informazioni presenti all'interno. Questo tipo di apprendimento perciò viene usato principalmente nel caso di reti feedforward.

Per addestrare una rete feedforward ed esser sicuri che abbia effettivamente compreso le relazioni ingresso-uscita, è necessario innanzitutto suddividere i dati per l'apprendimento in tre set. Il primo, il **training set**, è composto dai dati che l'algoritmo di addestramento userà direttamente per modificare i pesi della rete, deve essere sufficientemente ricco di informazioni da permettere alla rete di comprendere le relazioni ingresso-uscita e minimizzare l'errore tra l'uscita desiderata e la predizione della rete. Il secondo, il **validation set**, in base alla politica dell'apprendimento ed all'algoritmo utilizzato può essere utilizzato per migliorare l'apprendimento ed in generale a validarne i risultati. Il terzo, il **test set**, serve per verificare le capacità di generalizzazione della rete, ovvero se è in grado di fornire delle risposte soddisfacenti, in presenza di esempi per i quali la rete non è stata addestrata. Tipicamente un dataset è composto da un 60-70% di esempi per il training, 15-20% per la validazione ed un 15-20% per il test della rete.

Prima dell'addestramento della rete è necessario fare un certo pre/postprocessing dei dati di ingresso ed uscita per poter massimizzare l'apprendimento della rete. L'importanza del pre-processing dei dati diviene chiaro quando si pensa al fatto che molte reti multilayer feedforward utilizzano come funzione di trasferimento delle funzioni sigmoidi. Queste funzioni tendono a saturare in uscita per valori di attivazione non elevati. Se questo accade all'inizio del processo di addestramento, i gradienti saranno piuttosto piccoli e l'addestramento stesso risulterà piuttosto lento. Soprattutto nello strato d'ingresso, se i valori di input sono piuttosto ampi, i pesi di connessione verso il primo strato hidden saranno tipicamento piuttosto bassi per evitare l'effetto di saturazione dei neuroni. Per migliorare dunque l'apprendimento è una pratica standard normalizzare i dati di ingresso prima di applicarli alla rete. In generale, il processo di normalizzazione viene applicato sia ai vettori di dati in ingresso che a quelli di uscita del dataset. In questo modo le uscite della rete ricadranno sempre in range normalizzati, che poi verrano riportate agli ordini di grandezza originali tramite post-processing.

Altri pre-processing comuni sono processi di rimozione di esempi ripetuti, per evitare che misure uguali vengano ripresentate più volte, oppure l'estrazione delle componenti principali dai vettori di ingresso, in modo tale da fornire alla rete un dataset di features minimo-ottimo per l'addestramento. Altra pratica comune è la normalizzazione dei dati in modo che abbiano media nulla e varianza unitaria.

Backpropagation

La regola più usata per aggiustare i pesi di un neurone è la *delta rule*, anche nota come *regola di Widrow-Hoff.* Se si fa riferimento al neurone presentato in figura 5.2.4, preso $x = (x_1, \ldots x_n)$ come ingresso fornito al neurone e t ed y rispettivamente, l'uscita desiderata e l'uscita neurale, l'errore δ è dato da



$$\delta = t - y. \tag{5.8}$$

La delta rule tabilisce che la variazione del generico peso Δw_i sia:

$$\Delta w_i = \eta \delta x_i \tag{5.9}$$

dove η è un numero reale compreso tra 0 e 1, anche detto learning rate, che determina la velocità di apprendimento del neurone. La delta rule modifica in maniera proporzionale all'errore solo i pesi delle connessioni che hanno contribuito all'errore ($x_i \neq 0$), al contrario non vengono modificati se $x_i = 0$, poiche non si conosce il suo contributo all'errore. Il nuovo valore dei pesi risulta essere:

$$w_i = w_i + \Delta w_i. \tag{5.10}$$

Per una rete composta da un solo strato, come in figura 5.2.4 con n ingressi e p uscite l'errore può essere definito in vari modi, tipicamente può essere il valore assoluto della differenza tra uscita desiderata e predizione, oppure l'errore quadratico medio. Siano t_j e o_j , rispettivamente, l'uscita desiderata e l'uscita effettiva del neurone j. L'errore E_k commesso dalla rete sull'esempio k può essere definito come:



e l'errore globale commesso dalla rete su tutto il training set di m esempi è:

$$E = \sum_{k=1}^{m} E_k.$$
 (5.12)

É facile capire che l'errore dipende dai pesi w_{ij} di ciascun neurone e quindi in generale sarà rappresentabile come una superficie in funzione dei pesi stessi. Se la funzione d'errore è di tipo quadratico qualcosa di simile ad un paraboloide nello spazio dei pesi, o più in generale un'ipersuperficie nello spazio dei pesi.

Per la delta rule l'errore alla uscita j e la variazione del valore dei pesi sono:

$$\delta_j = t_j - o_j$$
$$\Delta w_{ij} = \eta \delta_j x_i$$

con t_j l'uscita desiderata dal neurone j, o_j l'uscita effettiva, x_i il segnale proveniente dal neurone $i \in \Delta w_{ij}$ la variazione del peso sulla connessione da i a j.

Si dimostra che, aggiornando i pesi mediante la delta rule, l'apprendimento converge verso una configurazione dei pesi che minimizza l'errore quadratico globale. Per far ciò basta dimostrare che tale regola è riconducibile alla forma

$$\Delta w_{ij} = -\frac{\partial E}{\partial w_{ij}} \tag{5.13}$$

che farebbe variare i pesi in modo tale da favorire la diminuzione dell'errore. Infatti:

- Se *E* cresce all'aumentare di w_{ij} si ha $\frac{\partial E}{\partial w_{ij}} > 0$, e w_{ij} viene diminuito $(\Delta w_{ij} < 0)$ per contrastare la crescita di *E*.
- Se *E* decresce all'aumentare di w_{ij} si ha $\frac{\partial E}{\partial w_{ij}} < 0$, e w_{ij} viene aumentato $(\Delta w_{ij} > 0)$ per favorire la diminuzione di *E*.

Per semplicità, consideriamo un neurone lineare con uscita definita dalla:

$$o_j = \sum_i x_i w_{ij} \tag{5.14}$$

Se si esprime la derivata dell'errore rispetto ad un peso prodotto di due quantità, la prima che esprime la variazione dell'errore in funzione dell'uscita di un neurone, la seconda che riguarda la variazione dell'uscita rispetto ad un peso:

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial o_j} \frac{\partial o_j}{\partial w_{ij}} \tag{5.15}$$

dalla (5.11) e dalla definizione di δ_j otteniamo:

$$\frac{\partial E}{\partial o_j} = -\delta_j \tag{5.16}$$

Inoltre $\frac{\partial o_j}{\partial w_{ij}} = x_i$, e di conseguenza:

$$\frac{\partial E}{\partial w_{ij}} = \delta_j x_i \tag{5.17}$$

Inserendo il learning rate otteniamo esattamente la delta rule.



Figura 5.6: Superficie dell'errore in funzione dei pesi w_{ij} con A minimo relativo e B minimo globale, C punto di partenza dell'algoritmo di addestramento.

Con riferimento alla figura 5.6, osserviamo che il processo di apprendimento può essere interpretato come una discesa sulla superficie dell'errore, lungo la linea di massima pendenza individuata dal gradiente:

$$-\nabla E = \left(-\frac{\partial E}{\partial w_{ij}}\right)$$

Il learning rate η rappresenta la rapidità di discesa dell'errore lungo la superficie, ed è dunque importante sceglierlo opportunamente, in quanto un

valore troppo piccolo comporta un apprendimento lento, mentre un valore troppo elevato può provocare oscillazioni dell'errore attorno al minimo. Tipicamente η viene fissato a valori alti all'inizio dell'apprendimento e lo si fa diminuire man mano che si procede con i passi.

Nel caso delle reti multistrato si vuole adottare un meccanismo di aggiornamento dei pesi simile alla delta rule, in cui l'errore è calcolato come differenza tra l'uscita desiderata e quella effettiva di ciascun neurone, ma si riuscirebbe ad aggiornare solo i pesi relativi ai neuroni di uscita, ma non quelli degli strati hidden. Questo perchè dello strato d'uscita di conosce l'uscita desiderata, mentre niente si sa a proposito dell'uscita desiderata dai neuroni nascosti.

Questo problema è stato risolto con l'introduzione dell'algoritmo di **backpropagation**, che prevede di calcolare l'errore commesso da un neurone dell'ultimo strato nascosto propagando all'indietro l'errore calcolato sui neuroni di uscita collegati a tale neurone. Lo stesso procedimento viene poi ripetuto per tutti i neuroni degli strati precedenti fino ad addestrare tutti gli strati.

L'algoritmo di backpropagation prevede che, all'inizio dell'algoritmo i pesi vengano inizializzati con valori casuali e per ogni esempio del training set, i segnali viaggino dall'ingresso verso l'uscita al fine di calcolare la risposta dela rete. Nella seconda fase i segnali di errore vengono propagati all'indietro sulle stesse connessioni in cui nella prima fase hanno viaggiato gli ingressi, ma in senso contrario, dall'uscita verso l'ingresso, in modo tale che durante questa fase vengano modificati i pesi. L'algoritmo prevede che le funzioni di trasferimento dei neuroni siano derivabili e usa una generalizzazione della delta rule.

Si può esprimere la derivata dell'errore come segue:

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial o_j} \frac{\partial o_j}{\partial a_j} \frac{\partial a_j}{\partial w_{ij}}$$
(5.18)

dove a_j è l'attivazione della funzione di trasferimento del neurone j ovvero la somma pesata degli ingressi più il bias. Se si pone:

$$\delta_j = -\frac{\partial E}{\partial a_j} \tag{5.19}$$

possiamo riscrivere l'equazione come:

$$\delta_j = -\frac{\partial E}{\partial o_j} \frac{\partial o_j}{\partial a_j} \tag{5.20}$$

L'errore commesso sul k-esimo esempio del training è:

$$E_k = \frac{1}{2} \sum_j (t_j - o_j)^2 \tag{5.21}$$

la cui derivata rispetto all'uscita del neurone vale:

$$\frac{\partial E}{\partial o_j} = -(t_j - o_j) \tag{5.22}$$

Per la funzione di trasferimento f (tipicamente, la funzione logistica o la tangente iperbolica) l'uscita e la sua derivata rispetto all'attivazione valgono:

$$o_j = f(a_j)$$
$$\frac{\partial o_j}{\partial a_j} = f'(a_j)$$

da cui si può scrivere:

$$\delta_j = (t_j - o_j)f'(a_j)$$

L'attivazione del neurone a_j e la sua derivata rispetto ai pesi w_{ij} valgono:

$$a_j = \sum_i x_i w_{ij}$$
$$\frac{\partial a_j}{\partial w_{ij}} = x_i$$

per cui ritornando alla formula di partenza risulta:

$$\frac{\partial E}{\partial w_{ij}} = -(t_j - o_j)f'(a_j)x_i = -\delta_j x_i$$

Applicando quindi il learning rate all'algoritmo, la modifica dei pesi sulla base della delta rule generalizzata prende la forma:

$$\nabla w_{ij} = \eta \delta_j x_i \tag{5.23}$$

L'errore δ_j è calcolabile per un neurone di uscita ma non per quello hidden, perchè non se ne conosce l'uscita desiderata. Il neurone hidden può essere adattato in modo proporzionale al suo contributo all'errore sullo strato successivo (verso l'uscita della rete). L'errore commesso dal neurone dell'ultimo strato hidden può essere calcolato come somma degli errori commessi da tutti i neuroni di uscita collegati a tale neurone hidden. Il contributi di ciascuno di questi errori dipende sia dalla dimensione dell'errore commesso dal relativo neurone di uscita e sia dal peso sulla connessione tra il neurone hidden e il neurone di uscita. Un neurone di uscita con un grosso errore contribuisce in maniera notevole all'errore di ogni neurone hidden a cui è connesso con un peso elevato. Per un neurone hidden, l'errore è dato da:

$$\delta_j = f'(a_j) \sum_s \delta_s w_{js} \tag{5.24}$$

dove s è l'indice dei neuroni dello strato che trasmette all'indietro l'errore. Negli strati hidden è vantaggioso usare funzioni logistiche o tangenti iperboliche perchè le derivate di tali funzioni son facilmente calcolabili, come segue:

$$f(x) = \frac{1}{1 + e^{-x}} \qquad f'(x) = f(x)(1 - f(x))$$
$$f(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}} \qquad f'(x) = (1 - f(x))(1 + f(x))$$

L'algoritmo basato sulla delta rule generalizzata ha i problemi che son stati descritti in precedenza, principalmente legati al learning rate η che deve essere modulato con una politica intelligente. Le tecniche per risolvere i problemi di oscillazione dell'errore e di convergenza son principalmente due. La prima consiste nel variare η nel tempo, la seconda prevede di ridurre la probabilità di oscillazione dei pesi usando un termine α detto momentum che è una costante di proporzionalità alla precedente variazione dei pesi. La legge di apprendimento con momentum diventa dunque:

$$\nabla w_{ij}(n+1) = \eta \delta_j o_i + \alpha \nabla w_{ij}(n) \tag{5.25}$$

con la quale la variazione dei pesi per l'esempio n+1 dipende dalla variazione dei pesi per l'esempio n.

Esistono due modalità di applicazione dell'algoritmo di backpropagation: la modalità batch, e quella incrementale (on-line). Nella prima i pesi vengono aggiornati dopo aver presentato alla rete tutti gli esempi del training set, mentre nella seconta i pesi sono aggiornati dopo la presentazione di ogni esempio, cosa molto utile nel caso di utilizzo online dell'addestramento.

Dato un trainin set costituito da m esempi di input e uscite target (X_k, T_k) l'algoritmo di backpropagation può essere visto complessivamente come la serie dei passi che seguono:

- 1. Si inizializzano i pesi w_{ij} con valori casuali, in genere non troppo elevati;
- 2. in base alla modalità di applicazione dell'algoritmo (batch o incrementale) si presentano gli ingressi e si calcolano le uscite o_j di tutti i neuroni della rete;
- 3. date le uscite target desiderate T_k si calcono gli errori δ_j e la variazione dei pesi Δw_{ij} per ogni neurone delo strato di uscita:

$$\delta_j = (t_j - o_j)f'(a_j)$$

4. partendo dall'ultimo strato hidden e procedendo all'indietro, si calcola:

$$\delta_j = f'(a_j) \sum_s \delta_s w_{js}$$

5. per tutti gli strati si aggiornano i pesi tramite momentum:

$$\nabla w_{ij}(n+1) = \eta \delta_j o_i + \alpha \nabla w_{ij}(n)$$

- 6. se l'algoritmo viene applicato in modalità incrementale, si ripete il processo dal punto 2 finchè non si sono presentati tutti gli m esempi del training set
- 7. si calcola l'errore sul training set e se è al di sotto di una soglia prefissata l'algoritmo termina, altrimenti si ripete un intero ciclo di presentazione del training set

Un ciclo di presentazione degli esempi del training set è detto epoca.

I problemi dell'algoritmo di backpropagation sono essenzialmente tre. Il primo è legato al learning rate η precedentemente descritto , il secondo è dovuto al fatto che è essenzialmente un algoritmo lento che può finire intrappolato in un valore minimo di E sub-ottimo, visto che l'errore avrà una serie di minimi relativi. Per questo motivo si ha inoltre un terzo problema legato al valore iniziale dei pesi w_{ij} che può far capitare di cominciare l'algoritmo da un punto della superficie di errore già troppo vicino ad un minimo locale.

Sebbene le reti dinamiche possano essere addestrate utilizzando gli stessi algoritmi basati sul gradiente come per le reti statiche, le prestazioni su di queste possono essere molto differenti, ed il gradiente inoltre deve essere calcolato in modo più complesso. Basti considerare il fatto che i pesi hanno due effetti diversi sull'uscita della rete. Il primo è l'effetto diretto perchè un cambiamento del peso provoca un immediato cambiamento nell'uscita del passo temporale corrente (calcolabile utilizzando la backpropagation standard), il secondo è un effetto indiretto perchè alcuni degli ingressi allo strato j per un istante t-1, sono a loro volta funzione dei pesi. Per tenere in conto degli effetti indiretti, viene usata una versione di backpropagation dinamica per calcolare i gradienti, che è computazionalmente più pesante. Ci si aspetta che reti del genere per essere addestrate hanno bisogno di tempi particolarmente maggiori rispetto alle reti statiche. Oltre a questo l'algoritmo di addestramento ha più probabilità di rimanere intrappolato in minimi locali, e questo comporta la necessità di addestrare una stessa rete più volte per raggiungere risultati ottimali.

Gli algoritmi di addestramento proposti fino ad ora sono algoritmi a gradiente di discesa e soffrono di tutti i problemi di convergenza sopra descritti. L'algoritmo di **Marquardt-Levemberg** che segue, è un'approssimazione del metodo di newton e risulta essere il più rapido a convergere quando il numero di pesi complessivi della rete risulta essere inferiore a qualche migliaio. Supponiamo di dover addestrare una rete con l'obiettivo di minimizzare una funzione $V(\underline{w})$ rispetto ai pesi \underline{w} . Il Metodo di Newton per l'aggiornamento dei pesi risulta essere:

$$\Delta \underline{w} = -[\nabla^2 V(\underline{w})]^{-1} \nabla V(\underline{w}) \tag{5.26}$$

dove $\nabla^2 V(\underline{w})$ è la matrice Hessiana e $\nabla V(\underline{w})$ è il gradiente. Se assumiamo $V(\underline{w})$ come la somma degli errori quadratici:

$$V(\underline{w}) = \sum_{i=1}^{N} e_i^2(\underline{w})$$
(5.27)

si dimostra che:

$$\nabla V(\underline{w}) = J^T(\underline{w})\underline{e}(\underline{w}) \tag{5.28}$$

$$\nabla^2 V(\underline{w}) = J^T(\underline{w}) J(\underline{w}) + S(\underline{w})$$
(5.29)

dove $J(\underline{w})$ è la matrice Jacobiana rispetto ai pesi e

$$S(\underline{w}) = \sum_{i=1}^{N} e_i(\underline{w}) \nabla^2 e_i(\underline{w})$$
(5.30)

Per il metodo di Gauss-Newton si assume che $S(\underline{w}) \approx 0$ e perciò:

$$\Delta \underline{w} = [J^T(\underline{w})J(\underline{w})]^{-1}J^T(\underline{w})\underline{e}(\underline{w})$$
(5.31)

La modifica Marquardt-Levenberg all'algoritmo Gauss-Newton è:

$$\Delta \underline{w} = [J^T(\underline{w})J(\underline{w}) + \mu I]^{-1}J^T(\underline{w})\underline{e}(\underline{w})$$
(5.32)

Il parametro μ viene moltiplicato da un fattore β ogni volta che un passo porti ad un aumento di $V(\underline{w})$. Quando un passo riduce $V(\underline{w})$, μ viene diviso per il fattore β . Da notare che se μ risulta essere grande, l'algoritmo diventa diventa a ripida discesa, mentre per valori piccoli diventa di Gauss-Newton. I termini dello jacobiano vengono calcolati utilizzando l'algoritmo di backpropagation standard.

Overfitting e capacità di generalizzazione

Una parte molto importante nello studio delle reti neurali e del loro apprendimento è quella che riguarda le capacità di generalizzazione. La rete deve comprendere il modello statistico dei dati, non deve limitarsi a memorizzare i soli dati del training set, altrimenti si crea il fenomeno detto overfitting. Più precisamente si parla di overfitting, ovvero eccessivo adattamento, quando un modelo statistico si adatta ai dati osservati usando un numero eccessivo di parametri. Un modello sbagliato può adattarsi perfettamente se è abbastanza complesso rispetto alla quantità di dati disponibili. Tipicamente si assume che la rete neurale raggiungerà uno stato in cui sarà in grado di predire gli output per tutti gli altri esempi che ancora non ha visionato, e quindi che il modello di apprendimento sarà in grado di generalizzare. Tuttavia, soprattutto nei casi in cui l'apprendimento è stato effettuato troppo a lungo o dove c'era uno scarso numero di esempi, il modello potrebbe adattarsi a caratteristiche che sono specifiche solo del training set ma che non hanno riscontro nel resto dei casi. In presenza di overfitting le prestazioni (capacità di adattamento/previsione) sul training set aumenteranno, mentre le prestazioni sui dati di test e validazione non visionati saranno peggiori.

Nel caso delle reti neurali è necessario attuare particolari tecniche che indichino quando un ulteriore addestramento non porterebbe ad una migliore generalizzazione. Si può osservare che se il numero dei parametri della rete è molto più piccolo del numero di esempi di addestramento, la possibilità di overfitting è scarsa se non addirittura nulla. Quindi se è possibile si dimensiona il training set in maniera opportuna per prevenire il problema. Nei casi in cui il training set è invece non adeguato alle dimensioni dela rete si può usare un'altra tecnica chiamata early stopping (arresto anticipato).

Questo metodo prevede di suddividere il dataset a disposizione in tre sottoinsiemi precedentemente descritti: il training set, il validation set ed il test set. Il training set è usato per calcolare il gradiente dell'errore e aggiornare i pesi. Durante il processo di addestramento si calcolano gli errori di validazione. Tipicamente gli errori di training e validation decrescono contemporaneamente durante la prima fase di addestramento, quando inizia l'overfitting l'errore sul validation set inizia a salire. Quando le prestazioni della rete sui vettori di validazione non migliorano per max-fail epoche di seguito, l'addestramento viene interrotto e vengono restituiti i pesi corrispondenti al minimo dell'errore di validazione. L'errore sul test set non viene utilizzato durante la fase di addestramento. Tale errore viene invece usato per confrontare tra loro reti diverse, ma può comunque essere utile monitorare l'errore sul test set durante l'addestramento. Se l'errore sul test set e l'errore sul validation set raggiungono minimi in epoche particolarmente diverse, si può sospettare di non aver effettuato una suddivisione appropriata in tre sottoinsiemi dell'insieme iniziale dei dati.

In figura 5.7 è mostrato l'effetto dell'overfitting per una rete addestrata per l'approssimazione della funzione seno, con dati un pò rumorosi usati solo per il training. L'assenza di validation set, comporta che la rete approssimi perfettamente i punti di training ma che complessivamente abbia prestazioni piuttosto scadenti.



Figura 5.7: Esempio di problema di overfitting.

5.2.5 Condizioni e regole sul dataset e l'architettura delle reti

Come già detto la generalizzazione è influenzata da tre fattori. Il primo è la grandezza del training set e quanto sia rappresentativo del fenomeno d'interesse, il secondo è l'architettura della rete neurale e il terzo è la complessità del problema che si sta affrontando. Chiaramente non è possibile avere il controllo sull'ultimo fattore, ma si può far qualcosa sui primi due. Nel contesto degli altri due fattori, possiamo vedere il problema della generalizzazione da due diverse prospettive:

- L'architettura della rete è fissata (in accordo con la complessità del problema) ed il problema da risolvere è la determinazione della grandezza del training set per avere una buona generalizzazione;
- La grandezza del dataset è fissata e il problema di interesse è determinare la migliore architettura della rete neurale per ottenere una buona generalizzazione.

Entrambi i punti di vista sono validi a loro modo. Nella pratica sembra che quello che necessita per avere una buona generalizzazione è avere un training set di dimensione N che soddisfa la condizione:

$$N = O\left(\frac{W}{\varepsilon}\right) \tag{5.33}$$

dive W è il numero totale di parametri della rete, ε è la frazione di errore permessa nel test set ed $O(\cdot)$ è l'ordine di grandezza di ciò che racchiudono le parentesi.

Una rete feedforward addestrata con un algoritmo di backpropagation può essere vista come un mezzo capace di eseguire una mappatura input-output non lineare di natura generale. Supposti $m_0 e M = m_L$ rispettivamente, il numero di nodi d'ingresso (features d'ingresso) e il numero dei neuroni d'uscita della rete, si può dire che le relazioni input-output della rete definiscono una mappa da uno spazio euclideo d'ingresso m_0 -dimensionale verso uno spazio euclideo di uscita M-dimensionale che è infinitamente differenziabile se le funzioni di trasferimento dei neuroni lo sono a loro volta. Per valutare le capacità della rete feedforward multistrato dal punto di vista della mappatura input-output, è fondamentale la sequente domanda: Qual'è il minimo numero di strati hidden in una rete feedforward multistrato con una mappatura input-output che fornisce una realizzazione approssimata di ogni mappatura continua?

Teorema universale di approssimazione

La risposta alla domanda precedente è compresa nel teorema universale di approssimazione per una mappa non lineare input-output, che dice come segue:

Sia $\varphi(\cdot)$ una funzione non costante, limitata e continua crescente monotona. Sia I_{m_0} un ipercubo unitario m_0 -dimensionale $[0,1]^{m_0}$. Lo spazio delle funzioni continue su I_{m_0} è indicato con $C(I_{m_0})$. Data una funzione $f \in C(I_{m_0})$ e $\varepsilon > 0$, esiste un intero m_1 e degli insiemi di costanti reali α_i, b_i e w_{ij} dove $i = 1, \ldots, m_1$ e $j = 1, \ldots, m_0$, tale che possiamo definire

$$F(x_1, \dots, x_{m_0}) = \sum_{i=1}^{m_1} \alpha_i \varphi \left(\sum_{j=1}^{m_0} w_{ij} x_j + b_i \right)$$
(5.34)

come una realizzazione approssimata della funzione f tale che

$$|F(x_1,\ldots,x_{m_0})-f(x_1,\ldots,x_{m_0})|<\varepsilon$$

per tutti i $x_1, x_2, \ldots, x_{m_0}$ che giaciono nello spazio d'ingresso.

Come si può intuire, il teorema universale di approsimazione è direttamente applicabile ad una rete feedforward multistrato. Come prima cosa si può notare che le funzioni di trasferimento a tangente iperbolica usate come nonlinearità della rete neurale sono funzioni non costanti, limitate e crescenti in maniera monotona e perciò soddisfano le condizioni imposte per la funzione $\varphi(\cdot)$ e che l'equazione (5.34) rappresenta l'uscita di una rete feedforward descritta come segue:

- la rete ha m_0 nodi d'ingresso ed un singolo strato hidden comporto da m_1 neuroni con ingressi $x_1, x_2, \ldots, x_{m_0}$;
- il neurone *i*-esimo ha i pesi pari a w_i, \ldots, w_{m_0} e bias b_i ;
- l'uscita della rete neurale è una combinazione lineare delle uscite degli strati hidden con $\alpha_1, \ldots, \alpha_{m_1}$ i pesi dello strato di uscita.

È importante dire che il teorema universale di approssimazione è un teorema di esistenza, nel senso che da una giustificazione matematica per per l'approssimazione di una funzione continua arbitraria rispetto ad una rappresentazione esatta. L'equazione (5.34) generalizza semplicemente l'approssimazione tramite serie finite di Fourier. In definitiva il teorema dichiara che è sufficiente avere una rete feedforward con un solo strato hidden per calcolare un'uniforme approssimazione dell'errore ε con un dato training set rappresentato da un insieme di ingressi $x_1, x_2, \ldots, x_{m_0}$ e un'uscita desiderata target $f(x_1, \ldots, x_{m_0})$, tuttavia il teorema non dice che una rete ad un solo strato hidden è ottima nel senso del tempo di apprendimento, che sia semplice da implementare o più in generale sappia generalizzare e tantomeno non da indicazioni su come trovare la rete ottima.

Limiti sull'approssimazione degli errori

Barron nel 1993 [50] ha determinato le proprietà di approssimazione di una rete feedforward, assunto che la rete abbia un singolo strato hidden con funzioni sigmoidai e strato d'uscita lineare. La rete è addestrata usando l'algoritmo di backpropagation ed infine testata su nuovi dati. Durante l'addestramento la rete impara punti specifici della funzione target f in base al training set e realizza la funzione di approssimazione F definita nell'equazione (5.34). Quando la rete è esposta ad un test set che non è stato visionato da questa durante l'apprendimento, la funzione complessiva F della rete agisce come uno stimatore di nuovi punti della funzione target, tale che $F = \hat{f}$.

Per introdurre i risultati di Barron, è necessario fare una premessa matematica. La proprietà di liscezza della funzione target f è espressa in termini della sua rappresentazione di Fourier. Sia $\tilde{f}(\omega)$, la trasformata di Fourier multidimensionale della funzione $f(\mathbf{x})$, con $\mathbf{x} \in \mathbb{R}^{m_0}$, ed ω il vettore delle frequenze di dimensione $m_0 \times 1$. La funzione f(x) in termini della sua trasformata di Fourier è definita dalla formula inversa:

$$f(x) = \int_{\mathbb{R}^{m_0}} \tilde{f}(\omega) e^{j\omega^T \mathbf{x}} d\omega$$
(5.35)

Possiamo definire il momento assoluto primo della distribuzione di ampiezza di Fourier come segue:

$$C_f = \int_{\mathbb{R}^{m_0}} |\tilde{f}(\omega)| \times \|\omega\|^{1/2} d\omega$$
(5.36)

dove $\|\omega\|$ è la norma euclidea del vettore riga delle pulsazioni. La media della norma delle frequenze pesate per l'ampiezza della trasformata di fourier è usata come misura dei limiti tra i quali oscilla la funzione f. In particolare C_f da una misura quantitativa della liscezza della funzione f ed oltre a questo da le basi per valutare i limiti di errore per una rete multi-strato feedforward con la sua funzione di mappatura input-output $F(\mathbf{x})$. L'approssimazione dell'errore è misurata attraverso l'errore quadratico integrato rispetto ad una misura arbitraria di probabilità μ in una sfera $B_r = {\mathbf{x} : ||\mathbf{x}|| \le r}$ di raggio r > 0. Su queste basi Barron afferma il seguente enunciato:

Per ogni funzione continua f con momento primo C_f finito ed ogni $m_1 \geq 1$, esiste una combinazione lineare di funzioni sigmoidi F(x) definite dall'equazione (5.34) tale che, quando la funzione f è osservata per un insieme di N valori dei vettori d'ingresso \mathbf{x} che sono limitati a giacere all'interno di B_r , il risultato fornisce il seguente limite al rischio empirico:

$$\xi_{av}(N) = \frac{1}{N} \sum_{i=1}^{N} (f(\mathbf{x}_i) - F(\mathbf{x}_i))^2 \le \frac{C'_f}{m_1}$$
(5.37)

dove $C'_{f} = (2rC_{f})^{2}$.

Il risultato che Barron ha trovato per quanto riguarda i risultati di approssimazione dell'equazione (5.38), usati per esprimere i limiti sul rischio $\xi_{av}(N)$ per l'ultilizzo di una rete feedforward con m_0 nodi d'ingresso e m_1 neuroni hidden, è come segue:

$$\xi_{av}(N) \le O\left(\frac{C_f'}{m_1}\right) + O\left(\frac{m_0 m_1}{N} \log N\right)$$
(5.38)

I due termini nel limite del rischio $\xi_{av}(N)$ esprimono il compromesso tra le due richieste in conflitto tra loro per quanto riguarda la dimensione dello strato hidden:

- 1. Accuratezza della migliore approssimazione: per soddisfare questo requisito la dimensione dello strato hidden m_1 deve essere ampia in accordo con il teorema universale di approssimazione;
- 2. Accuratezza dell'adattamento dell'approssimazione ai dati di validazione: per soddisfare questo secondo requisito, è necessario usare bassi rapporti m_1/N . Se la dimensione N del training è fissata, la dimensione dello strato hidden m_1 dovrebbe essere tenuta bassa, soluzione in conflitto con il primo requisito.

L'equazione (5.38) permette di fare alcune considerazioni. In particolare, non è richiesto un numero di esempi esponenzialmente largo ed ampio nella dimensionalità del numero delle features d'ingresso m_0 per ottenere una stima accurata della funzione target, a condizione che il momento assoluto primo C_f sia finito, risultato che rende le reti feedforward un ottimo strumento di approssimazione universale nella pratica.

Un'altra considerazione importante è il fatto che le reti neurali feedforward hanno una velocità di convergenza del rischio minore rispetto ad altre tecniche di approssimazione basate su polinomi o classiche funzioni trigonometriche. Queste ultime hanno un grosso problema dovuto alla dimensionalità delle features di ingresso, chiamato anche maledizione della dimensiona*lità*. Il concetto della maledizione della dimensionalità è stato introdotto da Richard Bellman nel 1961, e in termini pratici dice che preso un dataset di ingressi con m_0 features e N, esempi la densità di campionamento è proporzionale a N^{1/m_0} . Sia f, la funzione che vogliamo approssimare e che giace su uno spazio m_0 -dimensionale dell'ingresso. La funzione f è arbitrariamente complessa e completamente sconosciuta, perciò si ha il bisogno di un dataset piuttosto denso per approssimarla bene. Sfortunatamente, è difficile trovare un dataset denso nel caso di funzioni a grandi dimensioni e da qui deriva la maledizione della dimensionalità. In particolare si ha una crescita esponenziale della complessità come risultato di un aumento della dimensionalità m_0 , che quindi porta ad un deterioramento della capacità di "riempimento dello spazio" con punti casuali "ben distribuiti" per spazi a grandi dimensioni.

Esistono solo due modi per mitigare il problema della maledizione della dimensionalità. Il primo è quello di incorporare le conoscenze a priori sulla funzione sconosciuta da approssimare, il secondo modo è quello di progettare la rete in modo da garantire la liscezza della funzione incognita all'aumentare della dimensionalità dell'ingresso.

Considerazioni pratiche

Il teorema di approssimazione universale è importante dal punto di vista teorico, perchè fornisce uno strumento matematico e le relative condizioni necessarie per poter utilizzare in maniera redditizia le reti feedforward con uno strato nascosto come classe di soluzioni approssimate. L'utilità di questo teorema sta nel fatto che ci garantisce di trovare una soluzione che effettivamente esiste, tuttavia, essendo solamente un teorema di esistenza, attualmente non specifica come costruire ed addestrare una rete feedforward con le proprietà di approssimazione dichiarate.

Il teorema di approssimazione universale assume che la funzione continua che si intende approssimare sia conosciuta, e che sia disponibile una rete neurale con lo strato nascosto di dimensioni illimitate per la sua approssimazione. Chiaramente entrambe le assunzioni sono violate nella gran parte delle applicazioni pratiche.

Il problema con le reti neurali feedforward con un solo strato nascosto è che i neuroni tendono ad interagire tra di loro globalmente. In situazioni complesse questa interazione rende difficoltoso ottenere un'approssimazione di un punto senza peggiorare quelle in altri punti. Una rete neurale con due o più strati nascosti rende il processo di approssimazione più gestibile, che nel particolare procede come segue:

- 1. Vengono estratte le **Features Locali** nel primo strato nascosto, in modo tale che alcuni neuroni del primo strato si specializzino per essere usati per partizionare lo spazio di ingresso in regioni. Contemporaneamente i neuroni rimanenti imparano le features locali che caratterizzano queste regioni;
- 2. Nel secondo strato nascosto vengono estratte le **Features Globali**, ed in particolare, un neurone del secondo strato nascosto combina le uscite dei neuroni del primo strato nascosto, operando in una particolare regione dello spazio degli ingressi, in modo tale da apprendere le caratteristiche globali di quella regione, e produrre uscite nulle per le altre.

Altre considerazioni pratiche riguardano l'utilità dell'early stopping per migliorare le performance di validazione. Una teoria statistica del fenomeno dell'overfitting presentato da Amari (1996) mette in guardia sull'utilizzo del metodo dell'early stopping per il training delle reti neurali. In particolare quando la dimensione del training set N è minore del numero di parametri liberi della rete W, si ottengono miglioramenti alle performance di generalizzazione. Lo studio ha dimostrato che il fenomeno dell'overfitting si presenta quando N < 30W, ed in questi casi è particolarmente indicato l'uso del metodo per fermare il training. Al contrario, nel caso in cui N > 30W, i miglioramenti delle performance di generalizzazione prodotte dall'uso dell'early stopping sono piuttosto bassi. In altre parole, l'addestramento della rete è sempre soddisfacente quando la dimensione del training set è grande, comparata al numero dei parametri liberi della rete da addestrare.

Per concludere il discorso sulla dimensione della rete neurale, degli strati e dei neuroni, è importante dire che una rete con dimensione minima ha meno probabilità di imparare relazioni inesistenti dovute ad artefatti o al rumore intrinseco nella misura dei dati di addestramento, e perciò è in grado di generalizzare meglio i nuovi dati. Una rete di dimensione ottima è perciò robusta al rumore e gli artefatti di misura.

5.3 Dataset Acquisiti

In questa sezione vengono descritti i dataset acquisiti, utili per l'identificazione della relazione che lega il segnale elettromiografico dei principali gruppi muscolari di spalla e gomito e la coppia articolare sviluppata durante le contrazioni muscolari. In particolare si soffermerà l'attenzione sul protocollo di misura, descrivendone vantaggi e svantaggi per quanto riguarda l'efficacia nell'addestramento delle reti neurali.

Son state eseguite due sessioni di misura in giornate differenti con lo stesso soggetto. Nella prima son stati acquisiti di dataset utili all'addestramento ed alla validazione del metodo di stima di coppia ai giunti, nella seconda son stati acquisiti i dataset utili al test del metodo stesso. In particolare il dataset acquisito nella prima sessione è stato suddiviso in dataset per l'addestramento e dataset per la validazione, in percentuale pari rispettivamente al 70% e 30%.

Durante ciascuna sessione di misura l'workspace di lavoro dell'arto è stato discretizzato. In particolare in ciascuna sessione son stati acquisiti 4 punti giacenti sul piano passante per l'articolazione di spalla, parallelo al piano sagittale del corpo. I dataset per l'addestramento e la validazione sono stati acquisi sui punti mostrati in figura 5.8(a), mentre i dataset per il test sono stati acquisiti sui punti mostrati in figura 5.8(b).

In particolare in ciascuna sessione di misura sono state acquisite due categorie di dataset differenti in base all'assistenza fornita dall'esoscheletro:

• la prima riguarda una situazione di misura nella quale il soggetto indossa l'esoscheletro con l'arto allineato con i link dello stesso in modo tale che il peso del braccio non venga scaricato sulla struttura del robot e dunque che i gruppi muscolari compensino la gravità senza alcun sup-



Figura 5.8: Punti acquisiti sul piano sagittale per il dataset di (a)addestramento/validazione, (b) test.

porto. I dataset acquisiti in questa modalità avranno l'abbreviazione **NoGC**;

• la seconda riguarda di misura nella quale il soggetto indossa l'esoscheletro con l'arto allineato con i link dello stesso in modo tale che il peso del braccio e dell'avambraccio vengano scaricati sulla struttura dello stesso tramite una cinghia con chiusura a velcro per sorreggere il braccio e l'appoggio del polso sulla guida circolare dell'ultimo giunto (q_5). I dataset acquisiti in questa modalità avranno l'abbreviazione **GC**.

Ciascun dataset è composto da due grandezze:

- i segnali sEMG di tutti i principali gruppi muscolari interessati dai movimenti nel piano sagittale del corpo, per l'articolazione della spalla e quella di gomito;
- le forze esercitate dalla mano sulla maniglia-sensore di forza del sistema L-Exos, durante le contrazioni muscolari.

Protocollo

Il segnale elettromiografico è stato acquisito da 1 soggetto sano volontario di sesso maschile, d'età pari a 29 anni. Lo scopo dello studio consiste nell'identificare la relazione che lega il segnale elettromiografico alla coppia sviluppata dai gruppi muscolari coinvolti dalle articolazioni del giunto di spalla e di gomito, per movimenti nel piano sagittale del corpo. Per far ciò si è scelto di eseguire delle contrazioni isometriche vincolate dal sistema esoscheletrico L-Exos, con controllo in posizione ad alta impedenza. Il soggetto è seduto con la parte superiore del corpo solidale con lo schienale della poltrona fisso rispetto al manipolatore esoscheletrico.

La procedura di misura è stata divisa in due fasi:

- una prima fase di rilassamento muscolare in posizione seduta, durante la quale vengono misurati il segnale ECG e quelli elettromiografici dei gruppi muscolari maggiormente soggetti al crosstalk dovuto al tracciato ECG, durante la quale viene identificato il filtro ottimo per l'eliminazione del disturbo;
- una seconda fase, durante la quale il soggetto deve raggiungere 4 punti giacenti sul piano passante per l'articolazione della spalla, parallelo a quello sagittale. Su ciascuno, per ognuna delle 4 direzioni ortogonali che la mano può eseguire lungo il piano sagittale (avanti, giù, su, sotto), viene eseguita una serie di 15 ripetizioni intervallate da circa 10 secondi di rilassamento muscolare. Le ripetizioni non sono vincolate nel tempo e nella forza massima.

Si è scelto di non eseguire una prova di massima contrazione muscolare volontaria (MVC). Tutti i segnali elettromiografici vengono perciò normalizzati per il loro relativo massimo livello elettromiografico riscontrato durante la sessione di misura.

5.4 Addestramento delle reti neurali per la predizione della coppia

Una volta raccolti di dati, si procede alla definizione della tipologia e della struttura della rete neurale che ha lo scopo di modellare la relazione dinamica non lineare tra gli esempi elettromiografici (in ingresso) e le coppie articolari (in uscita). La tipologia della rete scelta è di tipo TDNN (figura 5.9), capace di modellare relazioni dinamiche e non lineari tra gli ingressi e le uscite presentate durante la fase di addestramento. Si è scelto di addestrare separatamente due reti neurali per l'identificazione di coppia, una per il giunto spalla Net_s ed una per quella di gomito Net_g, in modo tale che ciascuna sia specializzata a riconoscere le contrazioni dei soli gruppi muscolari effettivamente interessati dai movimenti dell'articolazione studiata. Le funzioni di trasferimento dei neuroni degli strati interni nascosti sono state scelte come tangenti iperboliche definite come segue:

$$\tanh\left(x\right) = \frac{1 - e^{-2x}}{1 + e^{-2x}} \tag{5.39}$$
La funzione di trasferimento dei neuroni dello strato d'uscita invece è stata scelta lineare.

Per quanto riguarda invece la definizione della struttura interna della rete TDNN (numero di strati nascosti, numero di neuroni per ciascun strato nascosto), viene eseguito uno studio di ottimizzazione con l'obiettivo di minimizzare l'errore commesso dalle reti neurali, compatibilmente alla teoria dell'ottimizzazione dei parametri liberi introdotta nel paragrafo 5.2.5 ed al teorema di approssimazione universale, in modo tale da bilanciare le esigenze di minimizzazione degli errori sul training set e contemporanemante sulla validation set e quindi la capacità della rete di generalizzare. Durante lo studio di ottimizzazione vengono fatti variare sia il numero di pesi della rete W che il numero di strati nascosti n_h . La struttura ottima della rete neurale è stata scelta con l'obiettivo di minimizzare la mediana del valore della funzione obiettivo d'addestramento su 40 tentativi di addestramento al variare di ogni parametro dell'ottimizzazione (W e n_h). Per una rete feedforward, con n_f numero di features di ingresso, n_h numero di strati nascosti, N neuroni in ogni strato nascosto, $n_{\boldsymbol{u}}$ numero di uscite, il numero totale di pesi della rete è dato dalla relazione che segue:

$$W = n_f N + (n_h - 1)N^2 + n_h N + Nn_u + n_u$$
(5.40)

La relazione che lega il numero di pesi al numero di neuroni è quadratica, perciò il numero di neuroni N per ciascun strato hidden, al variare del numero di pesi sarà la radice positiva dell'equazione (5.40) arrotondata per eccesso. Il numero dei ritardi di ingresso n_D della TDNN non è stato oggetto di ottimizzazione. La scelta del loro numero è stata effettuata in base al tipo di finestratura utilizzata per modellare la relazione tra i segnali sEMG e le coppie, se ne parlerà nel dettaglio in seguito.

Le due reti neurali Net_s e Net_g sono state addestrate con dataset definiti come coppie ingresso/uscita:

- per la rete di spalla Net_s, l'ingresso I_s composto da features del segnale sEMG di PM, DA e DP; l'uscita O_s rappresentata dalle coppie di spalla τ_s sul piano sagittale al giunto q_2 dell'esoscheletro L-Exos;
- per la rete di gomito Net_g , l'ingresso I_g composto da features del segnale sEMG di BB e TB; l'uscita O_g rappresentata dalle coppie di gomito τ_g sul piano sagittale al giunto q_4 dell'esoscheletro L-Exos.

Per la stima delle coppie è stato deciso di non usare le posture del L-Exos (e dunque dell'arto) in quanto l'informazione è già inclusa nella componente tonica del segnale sEMG.



Figura 5.9: Struttura delle TDNN Net_s e Net_q con n_D ritardi in ingresso.

La feature scelta come valore di ingresso per le TDNN ed estratta dal segnale sEMG (precedentemente preprocessato) è il valore assoluto medio (MAV) definito come segue:

$$MAV_{j} = \frac{1}{W_{l}} \sum_{i=1}^{W_{l}} |x_{j}(i)|$$
(5.41)

dove W_l è la lunghezza della finestra, $x_j(i)$ è il valore dell' i-esimo campione del segnale sEMG del j-esimo canale.

L'addestramento è stato eseguito per due modalità di finestratura (figura 5.10), la prima per finestre contigue di segnale sEMG e la seconda per finestre sovrapposte. Le features del segnale elettromiografico son state calcolate per finestre di segnale lunghe 200ms ($W_l = 51$ campioni a 256 Hz), per trovare un compromesso tra l'ottimizzazione delle prestazioni della rete (l'errore di stima per la relazione sEMG-coppia/forza si abbassa all'aumentare della lunghezza della finestra [41]) e le esigenze di utilizzo realtime della predizione di coppia. Il numero di ritardi n_D dello strato d'ingresso delle TDNN è stato scelto in maniera differente per le due modalità. Per quanto riguarda la modalità a finestre contigue è stato scelto un numero di ritardi pari a 2 finestre ($n_D = 2$), perciò al tempo t si intende predirre l'uscita attuale di coppia in relazione delle tre features consecutive Ft(t-2), Ft(t-1), Ft(t). Per quanto riguarda



Figura 5.10: Modalità di finestratura dei segnali: in alto la tecnica delle finestre contigue, in basso la tecnica delle finestre sovrapposte.

la modalità a finestre sovrapposte, è stata scelto un tempo di sovrapposizione pari a 100ms (25 campioni a 256 Hz), per aumentare la del controllo senza aumentare vertiginosamente il numero di esempi per l'addestramento, ed ottimizzare la rapidità dell'addestramento. In questa modalità il numero di ritardi n_D dello strato d'ingresso delle TDNN è stato scelto pari a 4, con l'intenzione di predirre l'uscita attuale di coppia in funzione delle 5 features consecutive Ft(t-5), $Ft(t-4), \ldots$, Ft(t).

Per le reti addestrate con la tecnica delle finestre contigue, il k-esimo elemento del dataset del p-esimo punto spaziale, per le reti Net_s e Net_e è un

vettore di features definito come segue:

$$\mathbf{I}_{s}^{p}(k) = \begin{bmatrix} MAV_{PM}(k) \\ MAV_{DA}(k) \\ MAV_{DP}(k) \\ MAV_{PM}(k-1) \\ MAV_{DA}(k-1) \\ MAV_{DP}(k-1) \\ MAV_{DP}(k-2) \\ MAV_{DP}(k-2) \\ MAV_{DP}(k-2) \end{bmatrix}$$
(5.42)
$$\mathbf{I}_{e}^{p}(k) = \begin{bmatrix} MAV_{BB}(k) \\ MAV_{BB}(k) \\ MAV_{TB}(k) \\ MAV_{TB}(k-1) \\ MAV_{TB}(k-1) \\ MAV_{BB}(k-2) \\ MAV_{TB}(k-2) \end{bmatrix}$$
(5.43)

Per le reti addestrate con la tecnica delle finestre sovrapposte, il k-esimo elemento del dataset del p-esimo punto spaziale, per le reti Net_s e Net_e è un vettore di features definito come segue:

$$\mathbf{I}_{e}^{p}(k) = \begin{bmatrix} MAV_{PM}(k) \\ MAV_{DA}(k) \\ MAV_{DP}(k) \\ MAV_{DP}(k-1) \\ MAV_{DA}(k-1) \\ MAV_{DP}(k-1) \\ \dots \\ MAV_{DP}(k-4) \\ MAV_{DA}(k-4) \\ MAV_{DP}(k-4) \end{bmatrix}$$
(5.44)
$$\mathbf{I}_{e}^{p}(k) = \begin{bmatrix} MAV_{BB}(k) \\ MAV_{BB}(k-1) \\ MAV_{TB}(k-1) \\ \dots \\ MAV_{BB}(k-4) \\ MAV_{TB}(k-4) \end{bmatrix}$$
(5.45)

Le features per la prima finestra di segnale sono state inizializzate col valore nullo.

Per quanto riguarda i valori di coppia ai giunti $\tau_s = \tau_{q_2} = \tau_2 \ e \ \tau_g = \tau_{q_4} = \tau_4$, usate come uscita delle due TDNN, vengono calcolati a partire dalle forze esterne $F_e e$ esercitate dall'utente in condizioni isometriche sull'end-effector del L-Exos, misurate tramite il sensore di forza triassiale sull'end-effector. Le coppie vengono calcolate secondo la (5.46) tramite l'uso dello jacobiano del robot J.

$$\tau = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \\ \tau_4 \end{bmatrix} = J^T F_{ee}$$
(5.46)

La scelta di utilizzare la coppia ai giunti L-Exos, invece di stimare una coppia muscolare alle articolazioni, è motivata dal fatto che i giunti di spalla e di gomito dell'esoscheletro L-Exos hanno l'asse di rotazione parallelo a quelli di spalla e gomito del corpo umano, in più i link hanno la stessa lunghezza del braccio e dell'avambraccio umano.

In particolare, il k-esimo elemento di uscita dei training dataset del p-esimo punto spaziale $O_s^p(k) \in O_e^p(k)$, sono definiti nelle (5.47) e (5.48) come la media dei valori di coppia in una finestra temporale pari a 200 ms ($W_l = 51$ campioni a 256 Hz).

$$O_s(k) = \frac{1}{W_l} \sum_{i=1}^{W_l} \tau_2(i)$$
(5.47)

$$O_e(k) = \frac{1}{W_l} \sum_{i=1}^{W_l} \tau_4(i)$$
(5.48)

Tutti gli esempi prima di essere passati alla rete neurale vengono processati come segue:

- una prima fase di selezione degli esempi, dove vengono eliminati gli eventuali esempi uguali, così che la rete non riceva lo stesso ingresso più di una volta;
- tutti gli esempi vengono normalizzati per il massimo valore della serie temporale, in modo tale da sfruttare le potenzialità delle funzioni di trasferimento scelte per i neuroni, e dunque che queste non rischino di saturare per valori troppo alti in ingresso. Oltre a questo, l'importanza della normalizzazione è legata anche alla variabilità di ampiezza della contrazione tra i vari soggetti. In tal senso, la normalizzazione è utile per migliorare ancora una volta la generalizzazione della rete neurale, e per poterla utilizzare anche con soggetti differenti da quelli per cui è stata addestrata.

Il dataset per l'addestramento è stato suddiviso in due parti: il 70% di esempi per il training, ed il 30% per la validazione. La suddivisione tra il dataset di training e quello di validazione, viene eseguita con modalità random. Tipicamente le reti neurali addestrate tramite l'algoritmo di feedforward soffrono di un problema dovuto alla presentazione temporale degli esempi: gli ultimi esempi presentati son quelli che vengono appresi meglio dalla rete. La possibilità di fornire alla rete gli esempi in successione random, permette di minimizzare questo problema.

Per l'addestramento delle reti neurali è stato utilizzato l'algoritmo di backpropagation Levenberg-Marquardt, che per reti neurali di dimensioni moderate (che non superano un numero di pesi superiore a diverse migliaia) risulta essere il più veloce a convergere [51]. La funzione obiettivo che si cerca di minimizzare è il valore quadratico medio degli errori (MSE) di predizione rispetto alle uscite target. L'addestramento viene fermato in base ai criteri che seguono:

- il valore MSE è inferiore a 0.01 $[N^2m^2]$;
- vengono eseguite 1000 epoche di addestramento;
- minimo gradiente inferiore a 10^{-5} ;
- tramite l'earling-stopping con un numero di verifiche di validazione consecutive pari a 8.

L'algoritmo di backpropagation per l'addestramento della rete neurale, ci permette di utilizzare la modalità *batch* e presentare contemporaneamente tutti gli esempi per tutte le 4 posizioni spaziali (punti A,B,C e D) che compongono il dataset. Le reti neurali Net_s e Net_g verranno perciò addestrate con delle celle di valori di input ed output il cui elemento k-esimo è composto come segue:

$$I_{s}(k) = \left\{ \begin{bmatrix} \mathbf{I}_{s}^{A}(k) & \mathbf{I}_{s}^{B}(k) & \mathbf{I}_{s}^{C}(k) & \mathbf{I}_{s}^{D}(k) \end{bmatrix} \right\}$$

$$I_{g}(k) = \left\{ \begin{bmatrix} \mathbf{I}_{g}^{A}(k) & \mathbf{I}_{g}^{B}(k) & \mathbf{I}_{g}^{C}(k) & \mathbf{I}_{g}^{D}(k) \end{bmatrix} \right\}$$

$$O_{s}(k) = \left\{ \begin{bmatrix} \mathbf{O}_{s}^{A}(k) & \mathbf{O}_{s}^{B}(k) & \mathbf{O}_{s}^{C}(k) & \mathbf{O}_{s}^{D}(k) \end{bmatrix} \right\}$$

$$O_{g}(k) = \left\{ \begin{bmatrix} \mathbf{O}_{g}^{A}(k) & \mathbf{O}_{g}^{B}(k) & \mathbf{O}_{g}^{C}(k) & \mathbf{O}_{g}^{D}(k) \end{bmatrix} \right\}$$

5.4.1 Risultati sul dataset noGC

L'addestramento delle reti di gomito e di spalla viene eseguito con il dataset senza compensazione di gravità dell'arto, al variare dei parametri topologici dei pesi(numero di connessioni) e strati nascosti, in entrambe le modalità di finestratura (contigua e sovrapposta). Ciascuna architettura delle reti (caratterizzate da n_h strati nascosti e W numero di pesi) è stata addestrata 40 volte. L'architettura ottimale delle TDNN è stata scelta con l'obiettivo di minimizzare la mediana degli errori quadratici medi sul dataset di validazione sui 40 addestramenti. Nel caso di addestramento con finestre contigue, il numero di elementi del dataset è pari a circa 2000, mentre nel caso di finestre sovrapposte è pari a circa 4000. Il numero di pesi ottimale sarà dunque limitato, con limite superiore il numero di esempi presentati alla rete durante l'addestramento. Il numero di strati nascosti son stati variare tenendo conto del teorema di approssimazione universale.

Risultati ottimizzazione con finestre contigue

I risultati del'ottimizzazione, nel caso dell'addestramento delle reti con la modalità delle finestre contigue, sono mostrati in figura 5.11. I grafici mostrano l'andamento della mediana dell'errore commesso dalle TDNN in funzione della variazione del numero degli strati nascosti n_h e del numero dei pesi W. In alto viene mostrato l'errore sul training set e sul validation set per la rete di spalla Net_s (a sinistra) e di gomito Net_g (a destra). Le reti ottimali per spalla e gomito risultano essere rispettivamente:

- Net_s: $n_h = 2$, W = 200, 9 neuroni per strato nascosto per un totale di 19 neuroni (18 nascosti, 1 output);
- Net_g: $n_h = 2$, W = 100, 6 neuroni per strato nascosto per un totale di 13 neuroni (12 nascosti, 1 output).

Preso come indice di performance la radice quadrata della mediana (RSM) dell'errore di predizione calcolato sul validation set, i risultati ottenuti per le Net_s e Net_g sono rispettivamente:

$$RSM_{noGC_s} = 2.13[Nm]$$
$$RSM_{noGC_g} = 1.35[Nm]$$

Le prestazioni della rete ottimale son state infine verificate sul test dataset, acquisito in giornata differente rispetto al training dataset. La verifica è stata eseguita per tre punti spaziali uguali al dataset del training set, e per un punto intermedio (vedi figura 5.8(b)). Per valutare i risultati ottenuti gli indici di performance scelti sono descritti come segue:



Figura 5.11: Risultati dell'ottimizzazione dell'architettura delle TDNN per la predizione di coppia basata su sEMG con finestre contigue e dataset senza compensazione di gravità.

- la radice quadrata dell'errore quadratico medio sulla stima di coppia rispetto alla coppia target (RMSE),
- il coefficiente di correlazione tra la la coppia target (T) e la coppia stimata (y), definito come:

$$CC = \frac{\sum_{i=1}^{n} (T_i - \overline{T}_i)(y_i - \overline{y}_i)}{\sqrt{\sum_{i=1}^{n} (T_i - \overline{T}_i)^2} \sqrt{\sum_{i=1}^{n} (y_i - \overline{y}_i)^2}}$$
(5.49)

L'indice RMSE è stato valutato per ciascuna rete e punto *i* acquisito (RMSE^{*i*}_{*s*} per la spalla, RMSE^{*i*}_{*g*} per il gomito), ed infine per tutto il dataset di test acquisito (RMSE^{*T*}_{*s*} e RMSE^{*T*}_{*g*}), mentre il coefficiente di correlazione è stato valutato per tutto il dataset di test, rispettivamente CC_{*s*} e CC_{*g*} per la rete di spalla e gomito. I risultati ottenuti sono i seguenti:

$$\begin{split} \text{RMSE}_{s}^{A} &= 3.13 [Nm] \\ \text{RMSE}_{g}^{A} &= 2.35 [Nm] \\ \text{RMSE}_{s}^{B} &= 3.20 [Nm] \\ \text{RMSE}_{g}^{B} &= 1.66 [Nm] \\ \text{RMSE}_{g}^{C} &= 3.24 [Nm] \\ \text{RMSE}_{g}^{C} &= 1.29 [Nm] \\ \text{RMSE}_{g}^{D} &= 1.29 [Nm] \\ \text{RMSE}_{g}^{D} &= 1.74 [Nm] \\ \text{RMSE}_{g}^{T} &= 3.13 [Nm] \\ \text{RMSE}_{g}^{T} &= 1.81 [Nm] \\ \text{CC}_{s} &= 0.9 \\ \text{CC}_{g} &= 0.68 \end{split}$$

Di seguito son mostrati i grafici temporali delle stime di coppia per ciascun punto spaziale del dataset di test.

Risultati ottimizzazione con finestre sovrapposte

I risultati del'ottimizzazione, nel caso dell'addestramento delle reti con la modalità delle finestre sovrapposte, sono mostrati in figura 5.20. I grafici mostrano l'andamento della mediana dell'errore commesso dalle TDNN in



Figura 5.12: Coppia acquisita (τ) e stimata con la tecnica delle finestre contigue (T) per i giunti di spalla e gomito durante movimenti in avanti e indietro sul punto A.



Figura 5.13: Coppia acquisita (τ) e stimata con la tecnica delle finestre contigue (T) per i giunti di spalla e gomito durante movimenti verso l'alto e basso sul punto A.



Figura 5.14: Coppia acquisita (τ) e stimata con la tecnica delle finestre contigue (T) per i giunti di spalla e gomito durante movimenti in avanti e indietro sul punto B.



Figura 5.15: Coppia acquisita (τ) e stimata con la tecnica delle finestre contigue (T) per i giunti di spalla e gomito durante movimenti verso l'alto e basso sul punto B.



Figura 5.16: Coppia acquisita (τ) e stimata con la tecnica delle finestre contigue (T) per i giunti di spalla e gomito durante movimenti in avanti e indietro sul punto C.



Figura 5.17: Coppia acquisita (τ) e stimata con la tecnica delle finestre contigue (T) per i giunti di spalla e gomito durante movimenti verso l'alto e basso sul punto C.



Figura 5.18: Coppia acquisita (τ) e stimata con la tecnica delle finestre contigue (T) per i giunti di spalla e gomito durante movimenti in avanti e indietro sul punto D.



Figura 5.19: Coppia acquisita (τ) e stimata con la tecnica delle finestre contigue (T) per i giunti di spalla e gomito durante movimenti verso l'alto e basso sul punto D.

funzione della variazione del numero degli strati nascosti n_h e del numero dei pesi W. In alto viene mostrato l'errore sul training set e sul validation set per la rete di spalla Net_s (a sinistra) e di gomito Net_g (a destra). Le reti



Figura 5.20: Risultati dell'ottimizzazione dell'architettura delle TDNN per la predizione di coppia basata su sEMG con finestre sovrapposte e dataset senza compensazione di gravità.

ottimali per spalla e gomito risultano essere rispettivamente:

- Net_s: $n_h = 2$, W = 800, 21 neuroni per strato nascosto per un totale di 43 neuroni (42 nascosti, 1 output);
- Net_e: $n_h = 2$, W = 380, 14 neuroni per strato nascosto per un totale di 29 neuroni (28 nascosti, 1 output).

Preso come indice di performance la radice quadrata della mediana (RSM) dell'errore di predizione calcolato sul validation set, i risultati ottenuti per le $Net_s \in Net_g$ sono rispettivamente:

•

$$RSM_{noGC_s} = 1.56[Nm]$$
$$RSM_{noGC_g} = 1.12[Nm]$$

Le prestazioni della rete ottimale son state infine verificate sul test dataset, acquisito in giornata differente rispetto al training dataset. La verifica è stata eseguita per tre punti spaziali uguali al dataset del training set, e per un punto intermedio (vedi figura 5.8(b)). Per valutare i risultati ottenuti gli indici di performance scelti sono descritti come segue:

- la radice quadrata dell'errore quadratico medio sulla stima di coppia rispetto alla coppia target (RMSE),
- il coefficiente di correlazione tra la la coppia target (T) e la coppia stimata (y), definito come:

$$CC = \frac{\sum_{i=1}^{n} (T_i - \overline{T}_i)(y_i - \overline{y}_i)}{\sqrt{\sum_{i=1}^{n} (T_i - \overline{T}_i)^2} \sqrt{\sum_{i=1}^{n} (y_i - \overline{y}_i)^2}}$$
(5.50)

L'indice RMSE è stato valutato per ciascuna rete e punto *i* acquisito (RMSE^{*i*}_{*s*} per la spalla, RMSE^{*i*}_{*g*} per il gomito), ed infine per tutto il dataset di test acquisito (RMSE^{*T*}_{*s*} e RMSE^{*T*}_{*g*}), mentre il coefficiente di correlazione è stato valutato per tutto il dataset di test, rispettivamente CC_s e CC_g per la

rete di spalla e gomito. I risultati ottenuti sono i seguenti:

$$\begin{split} \text{RMSE}_{s}^{A} &= 3.49[Nm] \\ \text{RMSE}_{g}^{A} &= 2.50[Nm] \\ \text{RMSE}_{g}^{B} &= 3.13[Nm] \\ \text{RMSE}_{g}^{B} &= 1.38[Nm] \\ \text{RMSE}_{g}^{C} &= 2.52[Nm] \\ \text{RMSE}_{g}^{C} &= 1.34[Nm] \\ \text{RMSE}_{g}^{D} &= 1.34[Nm] \\ \text{RMSE}_{g}^{D} &= 1.60[Nm] \\ \text{RMSE}_{g}^{T} &= 2.9[Nm] \\ \text{RMSE}_{g}^{T} &= 1.77[Nm] \\ \text{RMSE}_{g}^{T} &= 1.77[Nm] \\ \text{CC}_{s} &= 0.92 \\ \text{CC}_{g} &= 0.65 \end{split}$$

Di seguito son mostrati i grafici temporali delle stime di coppia per ciascun punto spaziale del dataset di test.



Figura 5.21: Coppia acquisita (τ) e stimata con la tecnica delle finestre sovrapposte (T) per i giunti di spalla e gomito durante movimenti in avanti e indietro sul punto A.



Figura 5.22: Coppia acquisita (τ) e stimata con la tecnica delle finestre sovrapposte (T) per i giunti di spalla e gomito durante movimenti verso l'alto e basso sul punto A.



Figura 5.23: Coppia acquisita (τ) e stimata con la tecnica delle finestre sovrapposte (T) per i giunti di spalla e gomito durante movimenti in avanti e indietro sul punto B.



Figura 5.24: Coppia acquisita (τ) e stimata con la tecnica delle finestre sovrapposte (T) per i giunti di spalla e gomito durante movimenti verso l'alto e basso sul punto B.



Figura 5.25: Coppia acquisita (τ) e stimata con la tecnica delle finestre sovrapposte (T) per i giunti di spalla e gomito durante movimenti in avanti e indietro sul punto C.



Figura 5.26: Coppia acquisita (τ) e stimata con la tecnica delle finestre sovrapposte (T) per i giunti di spalla e gomito durante movimenti verso l'alto e basso sul punto C.



Figura 5.27: Coppia acquisita (τ) e stimata con la tecnica delle finestre sovrapposte (T) per i giunti di spalla e gomito durante movimenti in avanti e indietro sul punto D.



Figura 5.28: Coppia acquisita (τ) e stimata con la tecnica delle finestre sovrapposte (T) per i giunti di spalla e gomito durante movimenti verso l'alto e basso sul punto D.

5.4.2 Risultati sul dataset con GC

L'addestramento delle reti di gomito e di spalla viene eseguito con il dataset con compensazione di gravità dell'arto, al variare dei parametri topologici dei pesi(numero di connessioni) e strati nascosti, in entrambe le modalità di finestratura (contigua e sovrapposta). Ciascuna architettura delle reti (caratterizzate da n_h strati nascosti e W numero di pesi) è stata addestrata 40 volte. L'architettura ottimale delle TDNN è stata scelta con l'obiettivo di minimizzare la mediana degli errori quadratici medi sul dataset di validazione sui 40 addestramenti. Nel caso di addestramento con finestre contigue, il numero di elementi del dataset è pari a circa 2800, mentre nel caso di finestre sovrapposte è pari a circa 5300. Il numero di pesi ottimale sarà dunque limitato, con limite superiore il numero di esempi presentati alla rete durante l'addestramento. Il numero di strati nascosti son stati variare tenendo conto del teorema di approssimazione universale.

Risultati ottimizzazione con finestre contigue

I risultati del'ottimizzazione, nel caso dell'addestramento delle reti con la modalità delle finestre contigue, sono mostrati in figura 5.29. I grafici mostrano l'andamento della mediana dell'errore commesso dalle TDNN in funzione della variazione del numero degli strati nascosti n_h e del numero dei pesi W. In alto viene mostrato l'errore sul training set e sul validation set per la rete di spalla Net_s (a sinistra) e di gomito Net_g (a destra). Le reti ottimali per



Figura 5.29: Risultati dell'ottimizzazione dell'architettura delle TDNN per la predizione di coppia basata su sEMG con finestre contigue e dataset con compensazione di gravità.

spalla e gomito risultano essere rispettivamente:

- Net_s: $n_h = 2$, W = 400, 15 neuroni per strato nascosto per un totale di 31 neuroni (30 nascosti, 1 output);
- Net_g: $n_h = 2$, W = 145, 8 neuroni per strato nascosto per un totale di 17 neuroni (16 nascosti, 1 output).

Preso come indice di performance la radice quadrata della mediana (RSM) dell'errore di predizione calcolato sul validation set, i risultati ottenuti per le $Net_s \in Net_g$ sono rispettivamente:

 $RSM_{GC_s} = 2.29[Nm]$ $RSM_{GC_g} = 1.79[Nm]$

Le prestazioni della rete ottimale son state infine verificate sul test dataset, acquisito in giornata differente rispetto al training dataset. La verifica è stata eseguita per tre punti spaziali uguali al dataset del training set, e per un punto intermedio (vedi figura 5.8(b)). Per valutare i risultati ottenuti gli indici di performance scelti sono descritti come segue:

- la radice quadrata dell'errore quadratico medio sulla stima di coppia rispetto alla coppia target (RMSE),
- il coefficiente di correlazione tra la la coppia target (T) e la coppia stimata (y), definito come:

$$CC = \frac{\sum_{i=1}^{n} (T_i - \overline{T}_i)(y_i - \overline{y}_i)}{\sqrt{\sum_{i=1}^{n} (T_i - \overline{T}_i)^2} \sqrt{\sum_{i=1}^{n} (y_i - \overline{y}_i)^2}}$$
(5.51)

L'indice RMSE è stato valutato per ciascuna rete e punto *i* acquisito (RMSE^{*i*}_{*s*} per la spalla, RMSE^{*i*}_{*g*} per il gomito), ed infine per tutto il dataset di test acquisito (RMSE^{*T*}_{*s*} e RMSE^{*T*}_{*g*}), mentre il coefficiente di correlazione è stato valutato per tutto il dataset di test, rispettivamente CC_{*s*} e CC_{*g*} per la

rete di spalla e gomito. I risultati ottenuti sono i seguenti:

$$\begin{aligned} \text{RMSE}_{s}^{A} &= 4.14[Nm] \\ \text{RMSE}_{g}^{A} &= 2.82[Nm] \\ \text{RMSE}_{s}^{B} &= 2.84[Nm] \\ \text{RMSE}_{g}^{B} &= 2.7[Nm] \\ \text{RMSE}_{s}^{C} &= 3.27[Nm] \\ \text{RMSE}_{s}^{C} &= 3.27[Nm] \\ \text{RMSE}_{g}^{D} &= 2.88[Nm] \\ \text{RMSE}_{g}^{D} &= 2.85[Nm] \\ \text{RMSE}_{g}^{D} &= 3.06[Nm] \\ \text{RMSE}_{g}^{T} &= 3.28[Nm] \\ \text{RMSE}_{g}^{T} &= 2.89[Nm] \\ \text{RMSE}_{g}^{T} &= 2.89[Nm] \\ \text{CC}_{s} &= 0.79 \\ \text{CC}_{g} &= 0.31 \end{aligned}$$

Di seguito son mostrati i grafici temporali delle stime di coppia per ciascun punto spaziale del dataset di test.



Figura 5.30: Coppia acquisita (τ) e stimata con la tecnica delle finestre contigue (T) per i giunti di spalla e gomito durante movimenti in avanti e indietro sul punto A.



Figura 5.31: Coppia acquisita (τ) e stimata con la tecnica delle finestre contigue (T) per i giunti di spalla e gomito durante movimenti verso l'alto e basso sul punto A.



Figura 5.32: Coppia acquisita (τ) e stimata con la tecnica delle finestre contigue (T) per i giunti di spalla e gomito durante movimenti in avanti e indietro sul punto B.



Figura 5.33: Coppia acquisita (τ) e stimata con la tecnica delle finestre contigue (T) per i giunti di spalla e gomito durante movimenti verso l'alto e basso sul punto B.



Figura 5.34: Coppia acquisita (τ) e stimata con la tecnica delle finestre contigue (T) per i giunti di spalla e gomito durante movimenti in avanti e indietro sul punto C.



Figura 5.35: Coppia acquisita (τ) e stimata con la tecnica delle finestre contigue (T) per i giunti di spalla e gomito durante movimenti verso l'alto e basso sul punto C.



Figura 5.36: Coppia acquisita (τ) e stimata con la tecnica delle finestre contigue (T) per i giunti di spalla e gomito durante movimenti in avanti e indietro sul punto D.



Figura 5.37: Coppia acquisita (τ) e stimata con la tecnica delle finestre contigue (T) per i giunti di spalla e gomito durante movimenti verso l'alto e basso sul punto D.

Risultati ottimizzazione con finestre sovrapposte

I risultati del'ottimizzazione, nel caso dell'addestramento delle reti con la modalità delle finestre sovrapposte, sono mostrati in figura 5.38. I grafici mostrano l'andamento della mediana dell'errore commesso dalle TDNN in funzione della variazione del numero degli strati nascosti n_h e del numero dei pesi W. In alto viene mostrato l'errore sul training set e sul validation set per la rete di spalla Net_s (a sinistra) e di gomito Net_g (a destra). Le reti ottimali per spalla e gomito risultano essere rispettivamente:

- Net_s: $n_h = 2$, W = 800, 21 neuroni per strato nascosto per un totale di 43 neuroni (42 nascosti, 1 output);
- Net_e: $n_h = 2$, W = 380, 14 neuroni per strato nascosto per un totale di 29 neuroni (28 nascosti, 1 output).

Preso come indice di performance la radice quadrata della mediana (RSM) dell'errore di predizione calcolato sul validation set, i risultati ottenuti per le Net_s e Net_g sono rispettivamente:

$$RSM_{GC_s} = 1.72[Nm]$$
$$RSM_{GC_g} = 1.57[Nm]$$



Figura 5.38: Risultati dell'ottimizzazione dell'architettura delle TDNN per la predizione di coppia basata su sEMG con finestre sovrapposte e dataset con compensazione di gravità.

Le prestazioni della rete ottimale son state infine verificate sul test dataset, acquisito in giornata differente rispetto al training dataset. La verifica è stata eseguita per tre punti spaziali uguali al dataset del training set, e per un punto intermedio (vedi figura 5.8(b)). Per valutare i risultati ottenuti gli indici di performance scelti sono descritti come segue:

- la radice quadrata dell'errore quadratico medio sulla stima di coppia rispetto alla coppia target (RMSE),
- il coefficiente di correlazione tra la la coppia target (T) e la coppia stimata (y), definito come:

$$CC = \frac{\sum_{i=1}^{n} (T_i - \overline{T}_i)(y_i - \overline{y}_i)}{\sqrt{\sum_{i=1}^{n} (T_i - \overline{T}_i)^2} \sqrt{\sum_{i=1}^{n} (y_i - \overline{y}_i)^2}}$$
(5.52)

L'indice RMSE è stato valutato per ciascuna rete e punto *i* acquisito (RMSE^{*i*}_{*s*} per la spalla, RMSE^{*i*}_{*g*} per il gomito), ed infine per tutto il dataset di test acquisito (RMSE^{*T*}_{*s*} e RMSE^{*T*}_{*g*}), mentre il coefficiente di correlazione è stato valutato per tutto il dataset di test, rispettivamente CC_s e CC_g per la rete di spalla e gomito. I risultati ottenuti sono i seguenti:

$$\begin{split} \text{RMSE}_{s}^{A} &= 3.87 [Nm] \\ \text{RMSE}_{g}^{A} &= 3.54 [Nm] \\ \text{RMSE}_{s}^{B} &= 3.97 [Nm] \\ \text{RMSE}_{g}^{B} &= 3.13 [Nm] \\ \text{RMSE}_{g}^{C} &= 3.61 [Nm] \\ \text{RMSE}_{g}^{C} &= 3.45 [Nm] \\ \text{RMSE}_{g}^{D} &= 3.72 [Nm] \\ \text{RMSE}_{g}^{T} &= 3.47 [Nm] \\ \text{RMSE}_{g}^{T} &= 3.5 [Nm] \\ \text{RMSE}_{g}^{T} &= 3.5 [Nm] \\ \text{CC}_{s} &= 0.77 \\ \text{CC}_{g} &= 0.17 \end{split}$$

Di seguito son mostrati i grafici temporali delle stime di coppia per ciascun punto spaziale del dataset di test.



Figura 5.39: Coppia acquisita (τ) e stimata con la tecnica delle finestre sovrapposte (T) per i giunti di spalla e gomito durante movimenti in avanti e indietro sul punto A.



Figura 5.40: Coppia acquisita (τ) e stimata con la tecnica delle finestre sovrapposte (T) per i giunti di spalla e gomito durante movimenti verso l'alto e basso sul punto A.



Figura 5.41: Coppia acquisita (τ) e stimata con la tecnica delle finestre sovrapposte (T) per i giunti di spalla e gomito durante movimenti in avanti e indietro sul punto B.



Figura 5.42: Coppia acquisita (τ) e stimata con la tecnica delle finestre sovrapposte (T) per i giunti di spalla e gomito durante movimenti verso l'alto e basso sul punto B.



Figura 5.43: Coppia acquisita (τ) e stimata con la tecnica delle finestre sovrapposte (T) per i giunti di spalla e gomito durante movimenti in avanti e indietro sul punto C.



Figura 5.44: Coppia acquisita (τ) e stimata con la tecnica delle finestre sovrapposte (T) per i giunti di spalla e gomito durante movimenti verso l'alto e basso sul punto C.



Figura 5.45: Coppia acquisita (τ) e stimata con la tecnica delle finestre sovrapposte (T) per i giunti di spalla e gomito durante movimenti in avanti e indietro sul punto D.



Figura 5.46: Coppia acquisita (τ) e stimata con la tecnica delle finestre sovrapposte (T) per i giunti di spalla e gomito durante movimenti verso l'alto e basso sul punto D.

5.4.3 Discussione ed analisi dei risultati

I risultati ottenuti sull'addestramento e l'ottimizzazione delle reti neurali con il dataset senza compensazione di gravità indicano la possibilità di poter predirre le coppie ai giunti con un buon valore dell'indice di performance sull'errore di validazione, sia per il giunto della spalla che per quello del gomito. In particolare è emerso, la possibilità di avere una stima migliore nel caso dell'utilizzo della tecnica delle finestre sovrapposte, rispetto all'uso della tecnica delle finestre contigue. Come ci si aspettava, i risultati mostrano la possibilità di ottenere una stima migliore sul giunto di gomito, ad un solo grado di libertà, attuato da meno gruppi muscolari rispetto al giunto articolare di spalla. I risultati sul dataset di test indicano la capacità della rete di spalla di avere un buon grado di generalizzazione sulle coppie esercitate durante contrazioni con forze con verso in avanti ed indietro, e un po' meno per le forze verso l'alto ed il basso. I risultati ottenuti sul punto di test B (vedi figura 5.8(b)), punto con dataset mai presentato alle reti durante l'addestramento, indicano la capacità della rete di spalla di buona capacità di generalizzazione. I risultati sul dataset di test indicano la capacità della rete di gomito di avere un buon grado di generalizzazione sulle coppie esercitate durante contrazioni con forze con verso rivolto verso l'alto ed il basso, e meno per le forze con verso in avanti e indietro. I risultati ottenuti sul punto di test B (vedi figura 5.8(b)), punto con dataset mai presentato alle reti durante l'addestramento, indicano la capacità della rete di gomito di avere una capacità di generalizzazione minore. I motivi che causano questa perdita di generalizzazione sono principalmente imputabili alla variazione delle condizioni fisiologiche tra una sessione di misura e l'altra, e soprattutto alla diversa posizione degli elettrodi tra una sessione e l'altra.

Si reputa necessario l'addestramento di una rete neurale di gomito durante una sessione di utilizzo del sistema di stima delle coppie, mentre per la stima della coppia di spalla può essere utilizzata la rete ottimale identificata durante lo studio.

I risultati ottenuti sull'addestramento e l'ottimizzazione della struttura delle reti nel caso di dataset con compensazione di gravità, mostrano invece una situazione più complessa. Le prestazioni di validazione ottenute sono peggiori rispetto al caso di addestramento con il dataset senza compensazione di gravità. Le reti di spalla sono in grado di conservare un discreto grado di generalizzazione, sia nel caso di finestratura contigua che nel caso di finestratura sovrapposta. Le reti di gomito invece non risultano essere in grado di generalizzare, sia nel caso di finestratura contigua che sovrapposta. I motivi che causano una prestazione peggiore sulla validazione sono sicuramente imputabili alla mancanza di una informazione di posizione all'interno del segnale elettromiografico, dovuto al fatto che la componente tonica del segnale sEMG (quella dovuta alla auto-compensazione della gravità dell'arto) è particolarmente bassa grazie al fatto che il peso dell'arto viene scaricato sulla struttura dell'esoscheletro. La componente tonica del segnale sEMG è quella relativa alla compensazione della gravità del braccio ed è strettamente correlata con la posizione dell'arto stesso (posizioni angolari del braccio rispetto alla gravità). La mancata capacità di generalizzare della rete di gomito, e la scarsa capacità di generalizzazione della rete di spalla, mostrate dai risultati sul dataset di test, sono dovute principalmente alla variazione di posizione dell'arto tra una sessione di misura e l'altra, e dunque alla variazione dello sgravo del peso dell'arto sulla struttura dell'esoscheletro. Questa variazione porta ad una variazione della componente tonica tra sessioni di misura differenti e dunque ad un diverso contenuto elettromiografico, non mappato durante la fase di addestramento.

In conclusione, il sistema ottimale per la stima delle coppie dell'esoscheletro L-Exos, attraverso i segnali sEMG, può essere implementato attraverso l'uso delle reti neurali TDNN, con la tecnica delle finestre sovrapposte, con una struttura a due strati nascosti ed un numero di pesi complessivi pari ad 800, per un numero di elementi di dataset d'addestramento pari a circa 4000. Il dataset con maggiori performance sulla validazione e sul test risulta essere quello acquisito nella situazione in cui il peso dell'arto viene bilanciato solo dalle contrazioni muscolari e non dalla struttura dell'esoscheletro.
Capitolo 6

Controllo in posizione e validazione

In questo capitolo verrà descritta nel dettaglio la catena di controllo in posizione del sistema esoscheletrico L-Exos, focalizzando l'attenzione su tutti i blocchi funzionali che la compongono, ed infine i risultati e le prestazioni ottenute.

6.1 Controllo in posizione

6.1.1 Schema di controllo

Il sistema di controllo implementato in questo lavoro di tesi è essenzialmente un controllo in posizione proporzionale derivativo (PD) sul riferimento di posizione derivato dal sistema di stima delle coppie descritto nel capitolo 5 con l'aggiunta della compensazione di gravità dei link del robot e dell'attrito viscoso. Il diagramma della catena di controllo è mostrato in figura 6.1 ed è composto da 5 moduli:

- il blocco di acquisizione dei segnali sEMG con lo stadio di pre-processamento dei segnali;
- il blocco di estrazione delle features dai segnali EMG;
- il di predizione delle coppie ai giunti;
- il blocco di generazione delle traiettorie ai giunti;
- il controllo in posizione PD con compensazione di gravità e attrito.



Figura 6.1: Flusso di controllo del sistema esoscheletrico L-Exos mediante elettromiografia di superficie.

Più in dettaglio, il blocco di acquisizione permette il campionamento ed il preprocessamento dei segnali EMG, campionati alla frequenza di 256 HZ e poi processati con filtro digitale passabanda 2-100 Hz e filtro notch centrato a 50Hz per eliminare la frequenza di rete. I muscoli del tronco, in questo caso il segnale sEMG del pettorale maggiore, sono interessati dal disturbo di crosstalk dell'ECG, vengono infine processati col blocco di rimozione del disturbo. Nel secondo blocco viene estratta la feature MAV dai segnali preprocessati, che vengono inviate all'ingresso del terzo blocco (Torque prediction), ne quale vengono stimate le coppie ai giunti 2 e 4 del L-Exos, rispettivamente indicati come T_s e T_g , con l'utilizzo delle TDNN Net_s e Net_g per spalla e gomito. Nel quarto modulo (Trajectory prediction), vengono stimate le posizioni dei giunti in base alle predizioni di coppia del blocco precedente. in particolare, per eliminare le oscillazioni di coppia attorno alla posizione di rest (coppia teorica nulla), viene utilizzata una *Deadband zone* in modo tale che le coppie siano nulla all'interno di una certa banda di soglia. Per ragioni di sicurezza, la coppia predetta viene saturata con un limite minimo ed uno massimo. Le coppie dei giunti processate vengono dunque trasformate in velocità dei giunti \dot{q}_2 e \dot{q}_4 , imponendo una relazione viscosa priva di massa tra le due entità, definita come:

$$\dot{q}_2 = K_s T_s \tag{6.1}$$

$$\dot{q}_4 = K_q T_q \tag{6.2}$$

dove le costanti di proporzionalità viscosa K_s e K_g sono scelte in modo che, per la massima coppia consentita, la velocità dei giunti sia pari a 0.5 rad/s. Le velocità così calcolate vengono dunque integrate per ottenere le posizioni di spalla e gomito nel piano sagittale ($q_2 e q_4$), limitate negli intervalli:

$$-11^{\circ} < q_2 < 28^{\circ} -90^{\circ} < q_4 < -64^{\circ}$$

Le posizioni dei giunti $q_1 \in q_3$ sono tenute fisse ai valori iniziali di riposo $q_{1\text{const}}$ e $q_{3\text{const}}$.

Le traiettorie di riferimento ottenute vengono inviate al controllore dell'esoscheletro L-Exos tramite ethernet con comunicazione UDP. Il controllo implementato a partire dalle traiettorie di riferimento è un controllo in posizione proporzonale derivativo (PD) ai giunti con compensazione di gravità e di attrito viscoso. In particolare il controllo in posizione è suddiviso in tre sottoblocchi:

1. il generatore di jerk (*jerk generator*), basato sulla tecnica di generazione della traiettoria OFSBJ proposta in [52]. Questo sottoblocco pianifica l'intera raiettoria a partire dalla conoscenza della traiettoria di riferimento, tenendo in conto i limiti meccanici di posizione, velocità, accelerazione e jerk dell' L-Exos. Il jerk viene quindi opportunamente integrato 3 volte per ottenere la traiettoria di riferimento da far seguire al controllo PD;

- 2. il calcolo delle componenti di coppia, durante il quale vengono calcolate le coppie dovute all'inseguimento della traiettoria ($\tau_P D$), alla compensazione della gravità dei link dell' L-Exos (GC, τ_G) e alla compensazione dell'attrito viscoso (VFC, τ_{VF});
- 3. il terzo sottoblocco, che somma le singole componenti di coppia per ottenere la coppia di riferimento τ_c ai giunti L-Exos.

Il sistema di controllo in posizione è implementato all'interno dell'unità di controllo che fornisce il supporto real time attraverso un sistema operativo real time. L'unità di controllo è basata su un pc industriale con un Single Board Computer (SBC). Il processore Pentium IV a 2.4MHz fornisce potenza di calcolo per avere una frequenza di controllo pari a 2KHz. Due schede I/O a 8 assi son presenti all'interno dell'unità di controllo. Il kernel del sistema di controllo è basato su un sistema operativo real-time che assicura alte capacità di performance (fino a 50kHz di tempo di commutazione, con pochi microsecondi di latenza). Tutti i dati I/O sono gestiti da specifici dispositivi realtime. La connessione con il PC remoto è basata sul protocollo UDP/IP che assicura la banda richiesta per i dati con basse latenze.

6.1.2 Risultati e validazione del controllo in posizione

I risultati mostrati in 5.4.1 e 5.4.2, hanno mostrato la necessità di dover predirre le coppie dei giunti di spalla e gomito a partire dai segnali sEMG nelle condizioni in cui le contrazioni muscolari sono utili anche al sostegno dell'arto superiore. Inoltre, risulta necessario l'addestramento delle due reti neurali (Net_s e Net_g) all'interno della stessa sessione di misura e utilizzo del sistema di controllo, in quanto le diverse condizioni fisiologiche dell'individuo e la diversa collocazione degli elettrodi durante due sessioni diverse di misura, possono portare ad una perdita di prestazioni del sistema di predizione della coppia a partire dai segnali sEMG. Il sistema di acquisizione ed il protocollo di misura ed addestramento delle reti neurali di spalla e gomito, permettono di ottenere un sistema di controllo sufficientemente accurato in un tempo di 20 minuti, con parametri addestrati sull'individuo.

È stata dunque eseguita una sessione di misura ed utilizzo del sistema di controllo, durante la quale, nella prima fase son stati acquisiti i segnali utili all'addestramento delle reti neurali, secondo il protocollo descritto in 5.3 e senza lo sgravio del peso dell'arto sulla struttura dell'esoscheletro, con i punti mostrati in figura 6.2, ed infine addestrate le reti neurali Net_s e Net_g secondo quanto descritto in 5.4, utilizzate infine per la predizione online delle coppie. Nella seconda fase, le TDNN addestrate vengono utilizzate nel sistema di controllo on-line del L-Exos mostrato in figura 6.1. Al soggetto utilizzatore è stato chiesto di eseguire movimenti continui sia lungo le quattro principali direzioni del piano sagittale, sia le posizioni intermedie nello stesso piano.



Figura 6.2: I quattro punti utilizzati per l'acquisizione del dataset in condizioni isometriche.

I risultati ottenuti durante il controllo on-line del sistema sono riportate in figura 6.3 per la spalla e in figura 6.4 per il gomito. In particolare son stati riportati:

- 1. i MAV estratti dai segnali sEMG nel tempo;
- 2. le coppie ai giunti acquisite (τ) e predette (T) nel tempo;
- 3. le posizioni di riferimento dei giunti (q) derivate dalle predizioni di coppia.

Gli indici di performance per la validazione del controllo proposto sono relativi all'errore di predizione di coppia per le TDNN addestrate in condizioni isometriche ed infine utilizzate per movimenti lenti. L'indice utilizzato per valutare la performance del controllo è la radice degli errori quadratici medi



Figura 6.3: Risultati del controllo basato su sEMG per il giunto spalla.



Figura 6.4: Risultati del controllo basato su sEMG per il giunto gomito.

(RSME) sulla predizione di coppia per la spalla (RSME_s) e per il gomito (RSME_g):

$$RMSE_s = 2.17[Nm]$$
$$RMSE_g = 1.19[Nm]$$

I risultati ottenuti per l'errore di predizione on-line della coppia dell'esoscheletro sono comparabili a quelli ottenuti per le TDNN addestrate in condizioni isometriche ($\text{RSM}_s = 1.56[Nm]$, $\text{RSM}_g = 1.12[Nm]$). Ne consegue che con il metodo proposto è possibile predirre le coppie ai giunti di spalla e gomito per lenti movimenti eseguiti sul piano sagittale. Questo risultato fa si che il sistema di controllo sia potenzialmente utilizzabile da pazienti con EMG residuo, durante la riabilitazione dell'arto superiore.

Capitolo 7

Conclusioni e lavori futuri

Il continuo aumento del numero di ictus con relative disabilità degli arti superiori sta portando all'aumento di importanza dello studio di metodi riabilitativi alternativi, per il recupero motorio dei pazienti colpiti da tali patologie croniche. In questo contesto, i sistemi robotici si stanno dimostrando strumenti efficaci per la creazione di terapie orientate a task motori ripetibili, con la possibilità di fornire stime oggettive dei progressi motori dei pazienti, riducendo sia i tempi di recupero che la necessità di personale specializzato per le sessioni di riabilitazione.

In questo lavoro è stato presentato un metodo di predizione delle coppie articolari di spalla e gomito, a partire dai segnali elettromiografici superficiali (sEMG), mediante l'utilizzo di un metodo model-free con reti neurali feedforward TDNN. In particolare è stato sviluppato un protocollo per l'acquisizione dei segnali sEMG dei principali gruppi muscolari di spalla e gomito e delle forze esercitate dalla mano lungo il piano sagittale del corpo e vincolate dall'esoscheletro L-Exos, durante contrazioni muscolari in condizioni isometriche in diversi punti del piano sagittale. È stato sviluppato lo schema di acquisizione dei segnali, per la gestione automatica di tutte le fasi del protocollo di misura e un'interfaccia grafica con le principali informazioni sui movimenti del protocollo. E stato inoltre sviluppato un sistema di cancellazione del disturbo di crosstalk ECG dai segnali sEMG. Son stati dunque acquisiti quattro dataset in due condizioni di lavoro differenti. Nel primo caso è stato chiesto al soggetto di sostenere da solo il peso dell'arto (dataset senza compensazione di gravità), nel secondo caso, il peso dell'arto è stato sostenuto dal sistema esoscheletrico L-Exos (dataset con compensazione di gravità). I dataset acquisiti son stati utilizzati per l'addestramento, la validazione ed il test delle TDNN. Durante l'addestramento è stato eseguito uno studio di ottimizzazione dei parametri topologici delle reti, in particolare sul numero di strati nascosti e sul numero di pesi complessivi delle reti. I risultati ottenuti

in questa fase di lavoro mostrano che le reti addestrate a partire dal dataset senza compensazione di gravità dell'arto permettono la stima delle coppie articolari con errori bassi e buone capacità di generalizzazione. La modalità di addestramento con finestre sovrapposte permette inoltre di ridurre gli errori di validazione e di ottenere migliori capacità di generalizzazione rispetto alla modalità a finestre contigue, oltre che una maggior frequenza di stima delle coppie. I risultati ottenuti durante l'addestramento a partire dal dataset con compensazione di gravità dell'arto risultano essere contrastanti. Da un lato, le reti di spalla risultano avere buone performance sull'errore di stima e medio-buone capacità di generalizzazione, peggiori rispetto a quelle addestrate con il dataset senza compensazione di gravità, dall'altro lato, le reti di gomito risultano avere scarse performance sull'errore di stima e mostrano di non avere buone capacità di generalizzazione. I risultati ottenuti sul dataset con compensazione di gravità hanno delle criticità dovute alla mancanza dell'informazione relativa alla posizione del braccio, contenuta nella componente tonica (di compensazione di gravità) del segnale elettromiografico, in questo caso quasi completamente annullata grazie al sostegno dell'esoscheletro. Ne consegue che il sistema di stima delle coppie alla spalla ed al gomito può essere costruito a partire da reti neurali TDNN addestrate con un dataset acquisito nella condizione in cui l'utente sostiene da solo il peso dell'arto superiore.

Nella seconda fase del lavoro di tesi, le TDNN addestrate sono state utilizzate per il controllo on-line del sistema esoscheletrico L-Exos. In particolare il controllo del robot è di tipo PD con compensazione di gravità e attrito viscoso, sui riferimenti di posizione calcolati in base alle predizioni di coppia delle TDNN. Durante la fase sperimentale, un soggetto ha utilizzato il sistema L-Exos con il controllo descritto implementato, ed in particolare gli è stato chiesto di eseguire dei movimenti continui lenti lungo il piano sagittale. Le reti TDNN (addestrate in condizioni isometriche) son risultate capaci di predirre le coppie ai giunti di spalla e gomito durante i movimenti continui lenti con valori degli indici di performance simili a quelli ottenuti durante la validazione ed il test delle stesse in condizioni isometriche. I risultati ottenuti promettono e dimostrano la possibilità di utilizzo dell'approccio di controllo sviluppato per il supporto dei movimenti dei pazienti durante la terapia.

A partire dai risultati ottenuti possono essere sviluppati diversi lavori futuri. In primis è necessario migliorare il sistema di cancellazione del disturbo di crosstalk ECG dai segnali sEMG dei muscoli del tronco, attraverso altri modelli di identificazione, oppure attraverso tecniche di rimozione con trasformate wavelet che permetterebbero di non utilizzare gli elettrodi per la misura del segnale ECG, e aumentare dunque la rapidità di vestizione degli elettrodi. Per quanto riguarda l'identificazione della relazione tra sEMG e coppia articolare possono essere percorse più strade. Nella prima è possibile migliorare il protocollo di acquisizione dei punti, in modo tale da minimizzare il numero di punti e numero di contrazioni isometriche per ciascun punto, utili per l'identificazione di un modello in grado d'avere un buon livello di generalizzazione. Sarebbe importante inoltre studiare approfonditamente la robustezza del metodo di predizione al variare della posizione degli elettrodi, in modo da poter ottenere un predittore di coppia utilizzabile in più sessioni di misura e terapia. Una seconda direzione di studio può incentrare l'attenzione sull'identificazione di un modello attraverso i segnali sEMG di contrazioni non isometriche e per movimenti complessi.

Un altro lavoro futuro è il test e l'eventuale adattamento del sistema di predizione delle coppie articolari su pazienti con EMG residuo e emiparesi dell'arto superiore, e l'utilizzo sugli stessi dell'approccio di controllo proposto.

Ulteriori lavori futuri potrebbero focalizzare l'attenzione sullo sviluppo di un controllo incentrato sull'identificazione dell'intenzione di movimento a partire dai segnali sEMG di pazienti con EMG residuo, in modo da creare un sistema di classificazione per il trigger dei movimenti dell'esoscheletro L-Exos per i pazienti non in grado di eseguire un movimento completo.

Bibliografia

- [1] Donald Lloyd-Jones, Robert J Adams, Todd M Brown, Mercedes Carnethon, Shifan Dai, Giovanni De Simone, T Bruce Ferguson, Earl Ford, Karen Furie, Cathleen Gillespie, et al. Executive summary: heart disease and stroke statistics-2010 update: a report from the american heart association. *Circulation*, 121(7):948, 2010.
- [2] D Inzitari. The italian guidelines for stroke prevention. the stroke prevention and educational awareness diffusion (spread) collaboration. Neurological sciences: official journal of the Italian Neurological Society and of the Italian Society of Clinical Neurophysiology, 21(1):5, 2000.
- [3] Sarah C Beckelhimer, Ann E Dalton, Charissa A Richter, Valerie Hermann, and Stephen J Page. Computer-based rhythm and timing training in severe, stroke-induced arm hemiparesis. *The American Journal* of Occupational Therapy, 65(1):96–100, 2011.
- [4] Mindy Lipson Aisen, H Igo Krebs, Neville Hogan, Fletcher McDowell, and Bruce T Volpe. The effect of robot-assisted therapy and rehabilitative training on motor recovery following stroke. Archives of Neurology, 54(4):443, 1997.
- [5] Peter S Lum, Charles G Burgar, Peggy C Shor, Matra Majmundar, and Machiel Van der Loos. Robot-assisted movement training compared with conventional therapy techniques for the rehabilitation of upperlimb motor function after stroke. Archives of physical medicine and rehabilitation, 83(7):952–959, 2002.
- [6] K. Englehart and B. Hudgins. A robust, real-time control scheme for multifunction myoelectric control. *Biomedical Engineering*, *IEEE Transactions on*, 50(7):848–854, 2003.
- [7] M.A. Oskoei and H. Hu. Support vector machine-based classification scheme for myoelectric control applied to upper limb. *Biomedical Engineering, IEEE Transactions on*, 55(8):1956–1965, 2008.

- [8] Z.O. Khokhar, Z.G. Xiao, C. Menon, et al. Surface emg pattern recognition for real-time control of a wrist exoskeleton. *Biomedical engineering online*, 9(1):41, 2010.
- [9] B. Hudgins, P. Parker, and R.N. Scott. A new strategy for multifunction myoelectric control. *Biomedical Engineering*, *IEEE Transactions on*, 40(1):82–94, 1993.
- [10] S. Micera, A.M. Sabatini, P. Dario, and B. Rossi. A hybrid approach to emg pattern analysis for classification of arm movements using statistical and fuzzy techniques. *Medical engineering & physics*, 21(5):303–311, 1999.
- [11] Y. Huang, K.B. Englehart, B. Hudgins, and A.D.C. Chan. A gaussian mixture model based classification scheme for myoelectric control of powered upper limb prostheses. *Biomedical Engineering, IEEE Transactions on*, 52(11):1801–1811, 2005.
- [12] Jack M Winters. Hill-based muscle models: a systems engineering perspective. In *Multiple muscle systems*, pages 69–93. Springer, 1990.
- [13] J.J. Luh, G.C. Chang, C.K. Cheng, J.S. Lai, and T.S. Kuo. Isokinetic elbow joint torques estimation from surface emg and joint kinematic data: using an artificial neural network model. *Journal of Electromyography* and Kinesiology, 9(3):173–183, 1999.
- [14] W. Ryu, B. Han, and J. Kim. Continuous position control of 1 dof manipulator using emg signals. In *Convergence and Hybrid Informa*tion Technology, 2008. ICCIT'08. Third International Conference on, volume 1, pages 870–874. IEEE, 2008.
- [15] C. Choi, S. Kwon, W. Park, HD Lee, and J. Kim. Real-time pinch force estimation by surface electromyography using an artificial neural network. *Medical engineering & physics*, 32(5):429, 2010.
- [16] S. Kwon and J. Kim. Real-time upper limb motion prediction from noninvasive biosignals for physical human-machine interactions. In Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on, pages 847–852. IEEE, 2009.
- [17] C.L. Pulliam, J.M. Lambrecht, and R.F. Kirsch. Electromyogram-based neural network control of transhumeral prostheses. J Rehabil Res Dev, 48:739–753, 2011.

- [18] K. Kiguchi and Y. Hayashi. An emg-based control for an upper-limb power-assist exoskeleton robot. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, 42(4):1064–1071, 2012.
- [19] P.K. Artemiadis and K.J. Kyriakopoulos. Emg-based control of a robot arm using low-dimensional embeddings. *Robotics, IEEE Transactions* on, 26(2):393–398, 2010.
- [20] Andrea d'Avella, Alessandro Portone, Laure Fernandez, and Francesco Lacquaniti. Control of fast-reaching movements by muscle synergy combinations. *The Journal of neuroscience*, 26(30):7791–7810, 2006.
- [21] Bokman Lim, Syungkwon Ra, and Frank C Park. Movement primitives, principal component analysis, and the efficient generation of natural motions. In *Robotics and Automation*, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on, pages 4630–4635. IEEE, 2005.
- [22] H.I. Krebs, N. Hogan, M.L. Aisen, and B.T. Volpe. Robot-aided neurorehabilitation. *Rehabilitation Engineering*, *IEEE Transactions on*, 6(1):75–87, 1998.
- [23] Hermano Igo Krebs, Bruce T Volpe, Dustin Williams, James Celestino, Steven K Charles, Daniel Lynch, and Neville Hogan. Robot-aided neurorehabilitation: a robot for wrist rehabilitation. Neural Systems and Rehabilitation Engineering, IEEE Transactions on, 15(3):327–335, 2007.
- [24] Laura Dipietro, Mark Ferraro, Jerome Joseph Palazzolo, Hermano Igo Krebs, Bruce T Volpe, and Neville Hogan. Customized interactive robotic treatment for stroke: Emg-triggered therapy. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, 13(3):325–334, 2005.
- [25] Charles G Burgar, Peter S Lum, Peggy C Shor, and HF Machiel Van der Loos. Development of robots for rehabilitation therapy: the palo alto va/stanford experience. *Journal of rehabilitation research and development*, 37(6):663–674, 2000.
- [26] Machiel Van der Loos. Robot-assisted movement training compared with conventional therapy techniques for the rehabilitation of upper-limb motor function after stroke. Arch Phys Med Rehabil, 83, 2002.
- [27] K. Kiguchi, S. Kariya, K. Watanabe, K. Izumi, and T. Fukuda. An exoskeletal robot for human elbow motion support-sensor fusion, adaptation, and control. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, 31(3):353–361, 2001.

- [28] K. Kiguchi, K. Iwami, M. Yasuda, K. Watanabe, and T. Fukuda. An exoskeletal robot for human shoulder joint motion assist. *Mechatronics*, *IEEE/ASME Transactions on*, 8(1):125–135, 2003.
- [29] K. Kiguchi, T. Tanaka, and T. Fukuda. Neuro-fuzzy control of a robotic exoskeleton with emg signals. *Fuzzy Systems, IEEE Transactions on*, 12(4):481–490, 2004.
- [30] R. Gopura, K. Kiguchi, and Y. Li. Sueful-7: a 7dof upper-limb exoskeleton robot with muscle-model-oriented emg-based control. In Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on, pages 1126–1131. IEEE, 2009.
- [31] T. Nef, M. Mihelj, G. Colombo, and R. Riener. Armin-robot for rehabilitation of the upper extremities. In *Robotics and Automation*, 2006. *ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 3152–3157. IEEE, 2006.
- [32] M. Mihelj, T. Nef, and R. Riener. Armin ii-7 dof rehabilitation robot: mechanics and kinematics. In *Robotics and Automation*, 2007 IEEE International Conference on, pages 4120–4125. IEEE, 2007.
- [33] Jacob Rosen, Moshe Brand, Moshe B Fuchs, and Mircea Arcan. A myosignal-based powered exoskeleton system. Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on, 31(3):210-222, 2001.
- [34] Ettore E Cavallaro, Jacob Rosen, Joel C Perry, and Stephen Burns. Real-time myoprocessors for a neural controlled powered exoskeleton arm. *Biomedical Engineering, IEEE Transactions on*, 53(11):2387–2396, 2006.
- [35] Joel C Perry, Jacob Rosen, and Stephen Burns. Upper-limb powered exoskeleton design. *Mechatronics*, *IEEE/ASME Transactions on*, 12(4):408–417, 2007.
- [36] B-C Tsai, W-W Wang, L-C Hsu, L-C Fu, and J-S Lai. An articulated rehabilitation robot for upper limb physiotherapy and training. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 1470–1475. IEEE, 2010.
- [37] Masatoshi Seki, Yuya Matsumoto, H Iijima, T Ando, Y Kobayashi, MG Fujie, and Masanori Nagaoka. Development of robotic upper limb orthosis with tremor suppressibility and elbow joint movability.

In Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on, pages 729–735. IEEE, 2011.

- [38] A. Jaskólska, K. Kisiel-Sajewicz, W. Brzenczek-Owczarzak, G.H. Yue, and A. Jaskólski. Emg and mmg of agonist and antagonist muscles as a function of age and joint angle. *Journal of Electromyography and Kinesiology*, 16(1):89–102, 2006.
- [39] P. Konrad. The abc of emg. A Practical Introduction to Kinesiological Electromyography, 1, 2005.
- [40] J. Hamill and K.M. Knutzen. Biomechanical Basis of Human Movement. Lippincott Williams & Wilkins.
- [41] D. Staudenmann, K. Roeleveld, D.F. Stegeman, and J.H. van Dieën. Methodological aspects of semg recordings for force estimation-a tutorial and review. *Journal of Electromyography and Kinesiology*, 20(3):375– 387, 2010.
- [42] PR Cavanagh and PV Komi. Electromechanical delay in human skeletal muscle under concentric and eccentric contractions. *European journal* Of applied physiology and occupational physiology, 42(3):159–163, 1979.
- [43] H.J. Hermens, B. Freriks, R. Merletti, D. Stegeman, J. Blok, G. Rau, C. Disselhorst-Klug, and G. Hägg. *European recommendations for surface electromyography*. Roessingh Research and Development The Netherlands, 1999.
- [44] J.D.M. Drake and J.P. Callaghan. Elimination of electrocardiogram contamination from electromyogram signals: An evaluation of currently used removal techniques. *Journal of electromyography and kinesiology*, 16(2):175–187, 2006.
- [45] P. Zhou, B. Lock, and T.A. Kuiken. Real time ecg artifact removal for myoelectric prosthesis control. *Physiological measurement*, 28(4):397, 2007.
- [46] Y. Hu, J. Mak, H. Liu, and K.D.K. Luk. Ecg cancellation for surface electromyography measurement using independent component analysis. In *Circuits and Systems, 2007. ISCAS 2007. IEEE International Symposium on*, pages 3235–3238. IEEE, 2007.
- [47] J. Taelman, B. Mijovic, S. Van Huffel, S. Devuyst, and T. Dutoit. Ecg artifact removal from surface emg signals by combining empirical mode

decomposition and independent component analysis. In Proc. of the International Conference on Bio-inspired Systems and Signal Processing, subconference of the 4th International Joint Conference on Biomedical Engineering Systems and Technologies (BIOSTEC 2011), 2010.

- [48] C. Marque, C. Bisch, R. Dantas, S. Elayoubi, V. Brosse, C. Perot, et al. Adaptive filtering for ecg rejection from surface emg recordings. *Journal* of electromyography and kinesiology: official journal of the International Society of Electrophysiological Kinesiology, 15(3):310, 2005.
- [49] G. Lu, J.S. Brittain, P. Holland, J. Yianni, A.L. Green, J.F. Stein, T.Z. Aziz, and S. Wang. Removing ecg noise from surface emg signals using adaptive filtering. *Neuroscience letters*, 462(1):14–19, 2009.
- [50] Simon S Haykin, Simon S Haykin, Simon S Haykin, and Simon S Haykin. Neural networks and learning machines, volume 3. Prentice Hall New York, 2009.
- [51] Martin T Hagan and Mohammad B Menhaj. Training feedforward networks with the marquardt algorithm. Neural Networks, IEEE Transactions on, 5(6):989–993, 1994.
- [52] Antonio Frisoli, Claudio Loconsole, Riccardo Bartalucci, and Massimo Bergamasco. A new bounded jerk on-line trajectory planning for mimicking human movements in robot-aided neurorehabilitation. *Robotics* and Autonomous Systems, 2012.
- [53] T.A. Kuiken, G. Li, B.A. Lock, R.D. Lipschutz, L.A. Miller, K.A. Stubblefield, and K.B. Englehart. Targeted muscle reinnervation for real-time myoelectric control of multifunction artificial arms. *JAMA: the journal* of the American Medical Association, 301(6):619–628, 2009.
- [54] R. Riener, T. Nef, and G. Colombo. Robot-aided neurorehabilitation of the upper extremities. *Medical and Biological Engineering and Computing*, 43(1):2–10, 2005.
- [55] R. Gopura, K. Kiguchi, and DSV Bandara. A brief review on upper extremity robotic exoskeleton systems. In *Industrial and Information* Systems (ICIIS), 2011 6th IEEE International Conference on, pages 346-351. IEEE, 2011.
- [56] S.K. Kundu and K. Kiguchi. Development of a 5 dof prosthetic arm for above elbow amputees. In *Mechatronics and Automation*, 2008. ICMA 2008. IEEE International Conference on, pages 207–212. IEEE, 2008.

- [57] K. Kiguchi and Y. Hayashi. Control of a myoelectric arm considering cooperated motion of elbow and shoulder joints. In *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*, pages 1616–1619. IEEE, 2011.
- [58] K. Kiguchi, M.H. Rahman, and M. Sasaki. Neuro-fuzzy based motion control of a robotic exoskeleton: considering end-effector force vectors. In *Robotics and Automation*, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on, pages 3146–3151. IEEE, 2006.
- [59] K. Kiguchi, Y. Imada, and M. Liyanage. Emg-based neuro-fuzzy control of a 4dof upper-limb power-assist exoskeleton. In *Engineering in Medicine and Biology Society*, 2007. EMBS 2007. 29th Annual International Conference of the IEEE, pages 3040–3043. IEEE, 2007.
- [60] R.A.R.C. Gopura and K. Kiguchi. Emg-based control of an exoskeleton robot for human forearm and wrist motion assist. In *Robotics and Au*tomation, 2008. ICRA 2008. IEEE International Conference on, pages 731–736. IEEE, 2008.
- [61] Antonio Frisoli, Fabrizio Rocchi, Simone Marcheschi, Andrea Dettori, Fabio Salsedo, and Massimo Bergamasco. A new force-feedback arm exoskeleton for haptic interaction in virtual environments. In Eurohaptics Conference, 2005 and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2005. World Haptics 2005. First Joint, pages 195–201. IEEE, 2005.
- [62] B. Katirji. Electromyography in Clinical Practice, A Case Study Approach. 2nd, edition, 2007.
- [63] M. Kutz, editor. Biomedical Engineering and Design Handbook: Biomedical Engineering Fundamentals. 2nd, edition, 2009.
- [64] M. Kutz, editor. Biomedical Engineering and Design Handbook: Biomedical Engineering Applications. 2nd, edition, 2009.
- [65] D. Knudson. Fundamentals of biomechanics. Fundamentals of Biomechanics, by Duane Knudson. Berlin: Springer, 2007. ISBN: 978-0-387-49311-4, 1, 2007.
- [66] Y. Makino, S. Ogawa, and H. Shinoda. Flexible emg sensor array for haptic interface. In SICE Annual Conference, 2008, pages 1468–1473. IEEE, 2008.

- [67] J.N.F. Mak, Y. Hu, and KDK Luk. Ica-based ecg removal from surface electromyography and its effect on low back pain assessment. In *Neural Engineering, 2007. CNE'07. 3rd International IEEE/EMBS Conference* on, pages 646–649. IEEE, 2007.
- [68] B. Rhou, M. Sawan, T. Desilets, and F. Bellemare. Real-time filtering technique to remove ecg interference from recorded esophageal emg. In *Biomedical Circuits and Systems Conference, 2008. BioCAS 2008. IEEE*, pages 21–24. IEEE, 2008.
- [69] M. Zardoshti-Kermani, B.C. Wheeler, K. Badie, and R.M. Hashemi. Emg feature evaluation for movement control of upper extremity prostheses. *Rehabilitation Engineering*, *IEEE Transactions on*, 3(4):324–333, 1995.
- [70] A. Phinyomark, P. Phukpattaranont, and C. Limsakul. Feature reduction and selection for emg signal classification. *Expert Systems with Applications*, 2012.
- [71] M.T. Hagan, H.B. Demuth, M.H. Beale, et al. Neural network design. Thomson Learning Stamford, CT, 1996.
- [72] M.M. Liu, W. Herzog, and H. Savelberg. Dynamic muscle force predictions from emg: an artificial neural network approach. *Journal of Electromyography and Kinesiology*, 9(6):391–400, 1999.
- [73] A. Erdemir, S. McLean, W. Herzog, A.J. van den Bogert, et al. Modelbased estimation of muscle forces exerted during movements. *Clinical Biomechanics (Bristol, Avon)*, 22(2):131–154, 2007.
- [74] O. Fukuda, T. Tsuji, A. Ohtsuka, and M. Kaneko. Emg-based humanrobot interface for rehabilitation aid. In *Robotics and Automation*, 1998. *Proceedings. 1998 IEEE International Conference on*, volume 4, pages 3492–3497. IEEE, 1998.
- [75] L.J. Hargrove, G. Li, K.B. Englehart, and B.S. Hudgins. Principal components analysis preprocessing for improved classification accuracies in pattern-recognition-based myoelectric control. *Biomedical Engineering*, *IEEE Transactions on*, 56(5):1407–1414, 2009.
- [76] E.A. Clancy and N. Hogan. Relating agonist-antagonist electromyograms to joint torque during isometric, quasi-isotonic, nonfatiguing contractions. *Biomedical Engineering*, *IEEE Transactions on*, 44(10):1024–1028, 1997.

- [77] T. Chau et al. A review of analytical techniques for gait data. part 2: neural network and wavelet methods. *Gait and Posture*, 13(2):102–120, 2001.
- [78] A. Zeghbib. Forearm surface EMG signals recognition and muscoloskeletal system dynamics identification using intelligent computational methods. 2007.
- [79] K. Manal, R.V. Gonzalez, D.G. Lloyd, and T.S. Buchanan. A real-time emg-driven virtual arm. *Computers in biology and medicine*, 32(1):25– 36, 2002.
- [80] P.K. Artemiadis and K.J. Kyriakopoulos. Estimating arm motion and force using emg signals: On the control of exoskeletons. In Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on, pages 279–284. IEEE, 2008.
- [81] P.K. Artemiadis and K.J. Kyriakopoulos. An emg-based robot control scheme robust to time-varying emg signal features. *Information Technology in Biomedicine*, *IEEE Transactions on*, 14(3):582–588, 2010.

Appendice A

Codice Matlab

In questo capitolo viene riportato il codice MATLAB sviluppato durante il lavoro di tesi.

A.1 Rimozione del crosstalk ECG

Codice per la stima della risposta impulsiva con ingresso ECG ed uscita disturbo di crosstak nell'emg a riposo.

A.1.1 estimateh.m

```
function h = estimateh2(emg,ecg,p)
% function h = estimateh2(emg,ecg,fs)
% Compute the impulsive response of tf "tf = emg(z)/ecg(z)"
%
% inputs:
% emg = EMG signals corrupted with ECG, low EMG activity
% ecg = simultaneously recorded ECG, size(ns,1)
% fs = sample frequency
% p = filter order
% output: h = impulsive response
    start = 1:
    stop = max(size(emg));
    % compute impulse response h
    nu = stop - start + 1;
    x = ecg(start:stop);
    U = zeros(nu,p); % matrix of delayed inputs
    U(:,1) = x;
for i = 2:p
        U(i:nu,i) = x(1:nu-i+1);
    end % for i
    y = emg(start:stop);
    h = U \setminus y;
```

A.1.2 ecgcorr.m

Funzione per la correzione del segnale EMG e l'eliminazione del disturbo di crosstalk dovuto all'ECG.

```
function out = ecgcorr(emg,ecg,h,fs)
% function out = ecgcorr(emg,ecg,fs)
\% inputs: emg = segnale EMG corrotto dal crosstalk ECG, bassa attività
% elettromiografica, size(ns,1)
% ecg = segnale ECG registrato simultaneamente, size(ns,1)
% h = risposta impulsiva del filtro di crosstalk ecg
% fs = frequenza di campionamento
% output: eout = emg corretto dal crosstalk ECG e filtrato passa-alto
[ Bhe Ahe ] = butter(4,20/(fs/2),'high');
[ns,nc] = size(emg);
ec = zeros(ns,nc);
for i = 1:nc
    ec1 = conv(ecg,h(:,i));
                                % apply convolution
    ec(:,i) = ec1(1:ns);
end
out = filter(Bhe,Ahe,emg-ec); % filtraggio passa-alto finale
```

A.2 Creazione Dataset

A.2.1 dataset_movimenti.m

```
= EMG(nell'ordine: PEC, DelAnt, DelPost, ECG, BB, TB)
% dati.Data(:,1:6)
% dati.Data(:,7:11)
                    = posizioni giunti(1,2,3,4,5)
% dati.Data(:,12:14) = forze F all'end-effector (X,Y,Z)
% dati.Data(:,15:18) = Coppie stimate tramite J'*F
% dati.Data(:,19)
                    = tempo xpc target
% dati.Data(:,20)
                    = punto del dataset (1 = punto A, 2 = punto B, etc..)
% dati.Data(:,21)
                    = stato(1 ECG, 2 REST, 3 avanti, 4 indietro, 5 su,
                       6 giù, 7 STOP)
                    = Numero di contrazioni
% dati.Data(:,22)
% dati.Data(:,23)
                    = rec_ecg (pari ad 1 se si è in fase di registrazione ECG)
                    = rec_rest (0 se non rest, 1 se rest_1, 2 se rest_2..etc)
% dati.Data(:,24)
% dati.Data(:,25)
                    = rec_forward (1 se il movimento è avanti, 0 altrimenti)
                    = rec_back (1 se il movimento è indietro, 0 altrimenti)
% dati.Data(:,26)
% dati.Data(:,27)
                    = rec_up (1 se il movimento è su, 0 altrimenti)
% dati.Data(:,28)
                     = rec_down (1 se il movimento è giù, 0 altrimenti)
pee = { 'A' 'B' 'C' 'D' };
mov_rest = {'forward' 'back' 'up' 'down' 'rest1' 'rest2' 'rest3' 'rest4'};
muscle = {'PectMajF' 'DelAnt' 'DelPost' 'BB' 'TB'};
                             % numero di punti nel dataset
      = size(pee,2);
np
     = size(muscle,2);
nmus
                              % numero di muscoli nel dataset
n_mov_rest =size(mov_rest,2); % numero di movimenti e rest
          = [find( dati.Data(i_all.A(1):i_all.A(2),24) == 1, 1, 'first'),...
i rest1.A
               find( dati.Data(i_all.A(1):i_all.A(2),24) == 1, 1, 'last')];
% indici di inizio-fine rest1
i_rest2.A = [find( dati.Data(i_all.A(1):i_all.A(2),24) == 2, 1, 'first'),...
               find( dati.Data(i_all.A(1):i_all.A(2),24) == 2, 1, 'last')];
% indici di inizio-fine rest1
i_rest3.A = [find( dati.Data(i_all.A(1):i_all.A(2),24) == 3, 1, 'first'),...
```

```
find( dati.Data(i_all.A(1):i_all.A(2),24) == 3, 1, 'last')];
% indici di inizio-fine rest1
i_rest4.A = [find( dati.Data(i_all.A(1):i_all.A(2),24) == 4, 1, 'first'),...
               find( dati.Data(i_all.A(1):i_all.A(2),24) == 4, 1, 'last')];
% indici di inizio-fine rest1
i_forward.A = [find( dati.Data(i_all.A(1):i_all.A(2),25) == 1, 1, 'first'),...
               find( dati.Data(i_all.A(1):i_all.A(2),25) == 1, 1, 'last')];
% indici di inizio-fine avanti
           = [find( dati.Data(i_all.A(1):i_all.A(2),26) == 1, 1, 'first'),...
i back.A
              find( dati.Data(i_all.A(1):i_all.A(2),26) == 1, 1, 'last')];
% indici di inizio-fine indietro
i_up.A
            = [find( dati.Data(i_all.A(1):i_all.A(2),27) == 1, 1, 'first'),...
              find( dati.Data(i_all.A(1):i_all.A(2),27) == 1, 1, 'last')];
% indici di inizio-fine su
i_down.A
            = [find( dati.Data(i_all.A(1):i_all.A(2),28) == 1, 1, 'first'),...
              find( dati.Data(i_all.A(1):i_all.A(2),28) == 1, 1, 'last')];
% indici di inizio-fine giù
i rest1.B
           = [find( dati.Data(i_all.B(1):i_all.B(2),24) == 1, 1, 'first'),...
               find( dati.Data(i_all.B(1):i_all.B(2),24) == 1, 1, 'last')];
% indici di inizio-fine rest1
i_rest2.B = [find( dati.Data(i_all.B(1):i_all.B(2),24) == 2, 1, 'first'),...
               find( dati.Data(i_all.B(1):i_all.B(2),24) == 2, 1, 'last')];
% indici di inizio-fine rest1
            = [find( dati.Data(i_all.B(1):i_all.B(2),24) == 3, 1, 'first'),...
i rest3.B
              find( dati.Data(i_all.B(1):i_all.B(2),24) == 3, 1, 'last')];
% indici di inizio-fine rest1
          = [find( dati.Data(i_all.B(1):i_all.B(2),24) == 4, 1, 'first'),...
i rest4.B
              find( dati.Data(i_all.B(1):i_all.B(2),24) == 4, 1, 'last')];
% indici di inizio-fine rest1
i_forward.B = [find( dati.Data(i_all.B(1):i_all.B(2),25) == 1, 1, 'first'),...
              find( dati.Data(i_all.B(1):i_all.B(2),25) == 1, 1, 'last')];
% indici di inizio-fine avanti
i_back.B
            = [find( dati.Data(i_all.B(1):i_all.B(2),26) == 1, 1, 'first'),...
              find( dati.Data(i_all.B(1):i_all.B(2),26) == 1, 1, 'last')];
% indici di inizio-fine indietro
            = [find( dati.Data(i_all.B(1):i_all.B(2),27) == 1, 1, 'first'),...
i_up.B
               find( dati.Data(i_all.B(1):i_all.B(2),27) == 1, 1, 'last')];
% indici di inizio-fine su
i_down.B
            = [find( dati.Data(i_all.B(1):i_all.B(2),28) == 1, 1, 'first'),...
              find( dati.Data(i_all.B(1):i_all.B(2),28) == 1, 1, 'last')];
% indici di inizio-fine giù
            = [find( dati.Data(i_all.C(1):i_all.C(2),24) == 1, 1, 'first'),...
i rest1.C
              find( dati.Data(i_all.C(1):i_all.C(2),24) == 1, 1, 'last')];
% indici di inizio-fine rest1
            = [find( dati.Data(i_all.C(1):i_all.C(2),24) == 2, 1, 'first'),...
i_rest2.C
              find( dati.Data(i_all.C(1):i_all.C(2),24) == 2, 1, 'last')];
% indici di inizio-fine rest1
i_rest3.C
            = [find( dati.Data(i_all.C(1):i_all.C(2),24) == 3, 1, 'first'),...
               find( dati.Data(i_all.C(1):i_all.C(2),24) == 3, 1, 'last')];
% indici di inizio-fine rest1
i_rest4.C = [find( dati.Data(i_all.C(1):i_all.C(2),24) == 4, 1, 'first'),...
              find( dati.Data(i_all.C(1):i_all.C(2),24) == 4, 1, 'last')];
% indici di inizio-fine rest1
i_forward.C = [find( dati.Data(i_all.C(1):i_all.C(2),25) == 1, 1, 'first'),...
               find( dati.Data(i_all.C(1):i_all.C(2),25) == 1, 1, 'last')];
% indici di inizio-fine avanti
i back.C
           = [find( dati.Data(i_all.C(1):i_all.C(2),26) == 1, 1, 'first'),...
              find( dati.Data(i_all.C(1):i_all.C(2),26) == 1, 1, 'last')];
% indici di inizio-fine indietro
           = [find( dati.Data(i_all.C(1):i_all.C(2),27) == 1, 1, 'first'),...
i_up.C
```

```
find( dati.Data(i_all.C(1):i_all.C(2),27) == 1, 1, 'last')];
% indici di inizio-fine su
           = [find( dati.Data(i_all.C(1):i_all.C(2),28) == 1, 1, 'first'),...
i down.C
              find( dati.Data(i_all.C(1):i_all.C(2),28) == 1, 1, 'last')];
% indici di inizio-fine giù
i_rest1.D
          = [find( dati.Data(i_all.D(1):i_all.D(2),24) == 1, 1, 'first'),...
              find( dati.Data(i_all.D(1):i_all.D(2),24) == 1, 1, 'last')];
% indici di inizio-fine rest1
          = [find( dati.Data(i_all.D(1):i_all.D(2),24) == 2, 1, 'first'),...
i_rest2.D
              find( dati.Data(i_all.D(1):i_all.D(2),24) == 2, 1, 'last')];
% indici di inizio-fine rest1
           = [find( dati.Data(i_all.D(1):i_all.D(2),24) == 3, 1, 'first'),...
i rest3.D
              find( dati.Data(i_all.D(1):i_all.D(2),24) == 3, 1, 'last')];
% indici di inizio-fine rest1
          = [find( dati.Data(i_all.D(1):i_all.D(2),24) == 4, 1, 'first'),...
i_rest4.D
              find( dati.Data(i_all.D(1):i_all.D(2),24) == 4, 1, 'last')];
% indici di inizio-fine rest1
i_forward.D = [find( dati.Data(i_all.D(1):i_all.D(2),25) == 1, 1, 'first'),...
              find( dati.Data(i_all.D(1):i_all.D(2),25) == 1, 1, 'last')];
% indici di inizio-fine avanti
           = [find( dati.Data(i_all.D(1):i_all.D(2),26) == 1, 1, 'first'),...
i back.D
              find( dati.Data(i_all.D(1):i_all.D(2),26) == 1, 1, 'last')];
% indici di inizio-fine indietro
           = [find( dati.Data(i_all.D(1):i_all.D(2),27) == 1, 1, 'first'),...
i_up.D
              find( dati.Data(i_all.D(1):i_all.D(2),27) == 1, 1, 'last')];
% indici di inizio-fine su
           = [find( dati.Data(i_all.D(1):i_all.D(2),28) == 1, 1, 'first'),...
i down.D
              find( dati.Data(i_all.D(1):i_all.D(2),28) == 1, 1, 'last')];
% indici di inizio-fine giù
for i=1:np
   for j=1:n_mov_rest
        eval(['emg.' pee{i} '.' mov_rest{j} '.muscles='...
              'emg.' pee{i} '.full_ds.muscles(:,i_' mov_rest{j} '.' pee{i}...
             '(1):i_' mov_rest{j} '.' pee{i} '(2));']);
       '(1):i_' mov_rest{j} '.' pee{i} '(2));']);
    end
```

```
end
```

A.3 Addestramento reti TDNN

A.3.1 meantorque.m

Funzione per il calcolo della coppia media per le finestre di segnale.

```
%
% [mav_signal,nf] = meantorque(torque,window_lenght,D)
% Calcola la media del torque, con l'algoritmo delle finestre sovrapposte
\% Le finestre di segnale son shiftate ognuna di un delay pari a D campioni.
if nargin < 3</pre>
    nf = floor(max(size(torque))/window_lenght);
    mean_T = zeros(min(size(torque)),nf);
    for i=1:nf
         mean_T(:,i) = mean(torque(:,(i-1)*window_lenght+1:i*window_lenght),2);
    end
else
    nf = floor((max(size(torque))-window_lenght)/D);
    mean_T = zeros(min(size(torque)),nf);
    for i = 1:nf
        mean_T(:,i) = mean(torque(:,(i-1)*D+1:i*D+window_lenght),2);
    end
end
```

A.3.2 MAV.m

Funzione per l'estrazione della feature MAV dai segnali sEMG.

```
function [mav_signal,nf_signal] = mav(signal,window_lenght,D)
% [mav_signal,nf_signal] = mav(signal,window_lenght)
%
% mav_signal = media dei valori assoluti del SIGNAL
% nf_signal = numero di finestre del SIGNAL
%
% calcola il mean absolute value del SIGNAL,
% per ogni finestra di WINDOW_LENGHT numero di campioni
%
% il SIGNAL deve essere un array di dimensioni n_canali x n_samples
\% la WINDOW_LENGHT deve essere la lunghezza della finestra in numero di
% campioni
%
% [mav_signal,nf_signal] = mav(signal,window_lenght,D)
% Calcola la MAV del signal, con l'algoritmo delle finestre sovrapposte
% Le finestre di segnale son shiftate ognuna di un delay pari a D campioni.
if nargin < 3</pre>
    nf = floor(max(size(signal))/window_lenght);
    mav_signal = zeros(min(size(signal)),nf);
    for i=1:nf
        mav_signal(:,i) = mean(abs(signal(:,(i-1)*window_lenght+1:i*window_lenght)),2);
    end
else
   nf = floor((max(size(signal))-window_lenght)/D);
   mav_signal = zeros(min(size(signal)),nf);
   for i = 1:nf
       mav_signal(:,i) = mean(abs(signal(:,(i-1)*D+1:i*D+window_lenght)),2);
   end
end
if nargout > 1
    nf_signal = nf;
end
```

A.3.3 dataset_e_addestramento.m

```
%% ADDESTRAMENTO DELLA RETE NEURALE
%% Settings
f = 256:
n_d = 0; % numero di campioni di ritardo
[Bh Ah] = butter(2,25/(f/2), 'low');
Of = 26; % ordine filtro ECG
% load dati registrati
load dati_GGMMAAAA.mat % formato data GG-MM-AAAA
filtro_ecg % calcolo del filtro ECG
% elenco dati:
% dati.Time
                    = tempo simulink
%
% dati.Data(:,1:6) = EMG(nell'ordine: PEC, DelAn
% dati.Data(:,7:11) = posizioni giunti(1,2,3,4,5)
                   = EMG(nell'ordine: PEC, DelAnt, DelPost, ECG, BB, TB)
% dati.Data(:,12:14) = forze F all'end-effector (X,Y,Z)
% dati.Data(:,15:18) = Coppie stimate tramite J'*F
% dati.Data(:,19)
                   = tempo xpc target
                   = punto del dataset (1 = punto A, 2 = punto B, etc..)
% dati.Data(:,20)
% dati.Data(:,21)
                   = stato(1 ECG, 2 REST, 3 avanti, 4 indietro, 5 su,
                      6 giù, 7 STOP)
                   = Numero di contrazioni
% dati.Data(:,22)
% dati.Data(:,23)
                   = rec_ecg (pari ad 1 se si è in fase di registrazione ECG)
% dati.Data(:,24)
                   = rec_rest (0 se non rest, 1 se rest_1, 2 se rest_2..etc)
                   = rec_forward (1 se il movimento è avanti, 0 altrimenti)
% dati.Data(:,25)
% dati.Data(:,26)
                   = rec_back (1 se il movimento è indietro, 0 altrimenti)
% dati.Data(:,27)
                   = rec_up (1 se il movimento è su, 0 altrimenti)
% dati.Data(:,28)
                   = rec_down (1 se il movimento è giù, 0 altrimenti)
%% DATT BACCOLTT
-0.219
       0.529]'
\% all'inizio del secondo punto (fatto così perchè il sistema automatico di
% cambio punto (se il Pee non viene raggiunto) fa passare il Pee prima da A
% e poi di nuovo verso il punto desiderato (algoritmo proxy dell'LExos)
i_start_A = find( dati.Data(:,20) == 1, 1, 'first');
i_end_A = find( dati.Data(:,20) == 1, 1, 'last');
t_smlnk.A = dati.Time( i_start_A:i_end_A -n_d); % Tempo di simulink
emg.A.full_ds.DelAnt =dati.Data(i_start_A:i_end_A-n_d,2);%raw emg Deltoid Anterior
emg.A.full_ds.DelPost=dati.Data(i_start_A:i_end_A-n_d,3);%raw emg Deltoid Posterior
emg.A.full_ds.ECG
                   =dati.Data(i_start_A:i_end_A-n_d,4);%raw emg ECG
emg.A.full_ds.BB
                   =dati.Data(i_start_A:i_end_A-n_d,5);%raw emg Biceps Brachii
emg.A.full_ds.TB
                   =dati.Data(i_start_A:i_end_A-n_d,6);%raw emg Triceps Brachii
%%%% filraggio emg pettorale
ecg_f_A = filter(Bh,Ah,emg.A.full_ds.ECG);
\% Correzione dal disturbo dell'ECG
emg.A.full_ds.PectMajF = ecgcorr(dati.Data(i_start_A:i_end_A-n_d,1),ecg_f_A,H,f);
emg.A.full_ds.muscles = [emg.A.full_ds.PectMajF, emg.A.full_ds.DelAnt,..
               emg.A.full_ds.DelPost, emg.A.full_ds.BB, emg.A.full_ds.TB]';
% coppie stimate tramite J'*F al secondo e quarto giunto (spalla, gomito)
dynamics.A.full_ds.torque = dati.Data(n_d+i_start_A:i_end_A,[16 18])';
clear ecg_f_A
-0.3, 0.6]
```

```
i_start_B = find( dati.Data(:,20) == 2, 1, 'first');
i_end_B = find( dati.Data(:,20) == 2, 1,'last');
t_smlnk.B = dati.Time( i_start_B:i_end_B -n_d); % Tempo di simulink
emg.B.full_ds.DelAnt =dati.Data(i_start_B:i_end_B-n_d,2);%raw emg Deltoid Anterior
emg.B.full_ds.DelPost =dati.Data(i_start_B:i_end_B-n_d,3);%raw emg Deltoid Posterior
emg.B.full_ds.ECG
                    =dati.Data(i_start_B:i_end_B-n_d,4);%raw emg ECG
                    =dati.Data(i_start_B:i_end_B-n_d,5);%raw emg Biceps Brachii
emg.B.full_ds.BB
emg.B.full_ds.TB
                    =dati.Data(i_start_B:i_end_B-n_d,6);%raw emg Triceps Brachii
%%%% filraggio emg pettorale
ecg_f_B = filter(Bh,Ah,emg.B.full_ds.ECG);
% Correzione dal disturbo dell'ECG
emg.B.full_ds.PectMajF = ecgcorr(dati.Data(i_start_B:i_end_B-n_d,1),ecg_f_B,H,f);
emg.B.full_ds.muscles = [emg.B.full_ds.PectMajF, emg.B.full_ds.DelAnt,...
               emg.B.full_ds.DelPost, emg.B.full_ds.BB, emg.B.full_ds.TB]';
% coppie stimate tramite J'*F al secondo e quarto giunto (spalla, gomito)
dynamics.B.full_ds.torque = dati.Data(n_d+i_start_B:i_end_B,[16 18])';
clear ecg_f_B
-0.219
       0.7],
i_start_C = find( dati.Data(:,20) == 3, 1,'first');
i_end_C = find( dati.Data(:,20) == 3, 1, 'last');
t_smlnk.C = dati.Time( i_start_C:i_end_C -n_d); % Tempo di simulink
emg.C.full_ds.DelAnt =dati.Data(i_start_C:i_end_C-n_d,2);%raw emg Deltoid Anterior
emg.C.full_ds.DelPost =dati.Data(i_start_C:i_end_C-n_d,3);%raw emg Deltoid Posterior
emg.C.full_ds.ECG
                 =dati.Data(i_start_C:i_end_C-n_d,4);%raw emg ECG
emg.C.full_ds.BB
                   =dati.Data(i_start_C:i_end_C-n_d,5);%raw emg Biceps Brachii
emg.C.full_ds.TB
                   =dati.Data(i_start_C:i_end_C-n_d,6);%raw emg Triceps Brachii
%%%% filraggio emg pettorale
ecg_f_C = filter(Bh, Ah, emg.C.full_ds.ECG);
% Correzione dal disturbo dell'ECG
emg.C.full_ds.PectMajF = ecgcorr(dati.Data(i_start_C:i_end_C-n_d,1),ecg_f_C,H,f);
emg.C.full_ds.muscles = [emg.C.full_ds.PectMajF, emg.C.full_ds.DelAnt,..
              emg.C.full_ds.DelPost, emg.C.full_ds.BB, emg.C.full_ds.TB]';
% coppie stimate tramite J'*F al secondo e quarto giunto (spalla, gomito)
dynamics.C.full_ds.torque = dati.Data(n_d+i_start_C:i_end_C,[16 18])';
clear ecg_f_C
-0.1 0.7]
i_start_D = find( dati.Data(:,20) == 4, 1, 'first');
i_end_D = find( dati.Data(:,20) == 4, 1, 'last');
t_smlnk.D = dati.Time( i_start_D:i_end_D -n_d); % Tempo di simulink
emg.D.full_ds.DelAnt =dati.Data(i_start_D:i_end_D-n_d,2);%raw emg Deltoid Anterior
emg.D.full_ds.DelPost =dati.Data(i_start_D:i_end_D-n_d,3);%raw emg Deltoid Posterior
                   =dati.Data(i_start_D:i_end_D-n_d,4);%raw emg ECG
emg.D.full_ds.ECG
emg.D.full_ds.BB
                   =dati.Data(i_start_D:i_end_D-n_d,5);%raw emg Biceps Brachii
emg.D.full_ds.TB
                   =dati.Data(i_start_D:i_end_D-n_d,6);%raw emg Triceps Brachii
%%%% filraggio emg pettorale
```

```
ecg_f_D = filter(Bh,Ah,emg.D.full_ds.ECG);
```

```
\% Correzione dal disturbo dell'ECG
emg.D.full_ds.PectMajF = ecgcorr(dati.Data(i_start_D:i_end_D-n_d,1),ecg_f_D,H,f);
emg.D.full_ds.muscles = [emg.D.full_ds.PectMajF, emg.D.full_ds.DelAnt,
                emg.D.full_ds.DelPost, emg.D.full_ds.BB, emg.D.full_ds.TB]';
% coppie stimate tramite J'*F al secondo e quarto giunto (spalla, gomito)
dynamics.D.full_ds.torque = dati.Data(n_d+i_start_D:i_end_D,[16 18])';
clear ecg_f_D
%% CALCOLO FEATURES
wl = 0.2;
                   % larghezza finestra (200 ms)
wd = 0.1;
                   % distanza tra due finestre sovrapposte
ncw = floor(f*wl); % numero di campioni della finestra
D = floor(f*wd); % numero di campioni di distanza tra due finestre sovrapposte
muscle = {'PectMajF' 'DelAnt' 'DelPost' 'BB' 'TB'};
pee = {'A' 'B' 'C' 'D'};
np = size(pee,2); %
                     % numero di punti nel dataset test
%% SELEZIONE SEGNALE UTILE
                               , 1, 'first')
i_data.A = [find(t_smlnk.A > 0)]
            find(t_smlnk.A < inf , 1, 'last')];</pre>
i_data.B = [find(t_smlnk.B > 0 , 1, 'first')
            find(t_smlnk.B < inf , 1, 'last')];</pre>
i_data.C = [find(t_smlnk.C > 0 , 1, 'first')
            find(t_smlnk.C < inf , 1, 'last')];</pre>
i_data.D = [find(t_smlnk.D > 0 , 1, 'first')
            find(t_smlnk.D < inf , 1, 'last')];</pre>
%%%%%%%%%%% dataset input EMG
input.A = emg.A.full_ds.muscles(:,i_data.A(1):i_data.A(2)) ;
input.B = emg.B.full_ds.muscles(:,i_data.B(1):i_data.B(2)) ;
input.C = emg.C.full_ds.muscles(:,i_data.C(1):i_data.C(2)) ;
input.D = emg.D.full_ds.muscles(:,i_data.D(1):i_data.D(2)) ;
%%%%%%%%%%% dataset target LExos %coppia ai giunti= Jt*F
output.A = dynamics.A.full_ds.torque(:,i_data.A(1):i_data.A(2));
output.B = dynamics.B.full_ds.torque(:,i_data.B(1):i_data.B(2));
output.C = dynamics.C.full_ds.torque(:,i_data.C(1):i_data.C(2));
output.D = dynamics.D.full_ds.torque(:,i_data.D(1):i_data.D(2));
for i=1:np
    % CALCOLO MAV e COPPIA MEDIA CON FINESTRE SOVRAPPOSTE
    eval(['[mav_emg_ovrlpd.' pee{i} ',nf_ovrlpd.' pee{i} ']=mav(input.'...
        pee{i} ',ncw,D);']);
    eval(['mean_torque_ovrlpd.' pee{i} '=meantorque(output.' pee{i} ',ncw,D);']);
    % CALCOLO MAV e COPPIA MEDIA CON FINESTRE CONTIGUE
    eval(['[mav_emg.' pee{i} ',nf.' pee{i} ']=mav(input.' pee{i} ',ncw);']);
    eval(['mean_torque.' pee{i} '=_meantorque(output.' pee{i} ',ncw);']);
end
%% RIMOZIONE OFFSET dalla coppia MEDIA dovuto all'offset del sensore di forza
\% dal dataset dei 4 punti si calcola la media del primo tratto di rest per
% i 4 punti che successivamente si sottrae
%indici di inizio-fine rest1 punto A
            = [find( dati.Data(i_start_A:i_end_A,24) == 1, 1, 'first'),...
i_rest1.A
               find( dati.Data(i_start_A:i_end_A,24) == 1, 1, 'last')];
% indici di inizio-fine rest1 punto B
```

```
= [find( dati.Data(i_start_B:i_end_B,24) == 1, 1, 'first'),...
i rest1.B
                find( dati.Data(i_start_B:i_end_B,24) == 1, 1, 'last')];
% indici di inizio-fine rest1 punto C
            = [find( dati.Data(i_start_C:i_end_C,24) == 1, 1, 'first'),...
i rest1.C
                find( dati.Data(i_start_C:i_end_C,24) == 1, 1, 'last')];
% indici di inizio-fine rest1 punto D
i_rest1.D
            = [find( dati.Data(i_start_D:i_end_D,24) == 1, 1, 'first'),...
                find( dati.Data(i_start_D:i_end_D,24) == 1, 1, 'last')];
offset.A = mean(meantorque(dynamics.A.full_ds.torque(:,i_rest1.A(1):...
                                  i_rest1.A(2)),ncw), 2 );
offset.B = mean(meantorque(dynamics.B.full_ds.torque(:,i_rest1.B(1):...
                                  i_rest1.B(2)),ncw), 2 );
offset.C = mean(meantorque(dynamics.C.full_ds.torque(:,i_rest1.C(1):...
                                  i_rest1.C(2)),ncw), 2 );
offset.D = mean(meantorque(dynamics.D.full_ds.torque(:,i_rest1.D(1):...
                                  i_rest1.D(2)),ncw), 2 );
offset_o.A = mean(meantorque(dynamics.A.full_ds.torque(:,i_rest1.A(1):...
                                  i_rest1.A(2)),ncw,D), 2 );
offset_o.B = mean(meantorque(dynamics.B.full_ds.torque(:,i_rest1.B(1):...
                                  i_rest1.B(2)),ncw,D), 2 );
offset_o.C = mean(meantorque(dynamics.C.full_ds.torque(:,i_rest1.C(1):...
                                  i_rest1.C(2)),ncw,D), 2 );
offset_o.D = mean(meantorque(dynamics.D.full_ds.torque(:,i_rest1.D(1):...
                                  i_rest1.D(2)),ncw,D), 2 );
mean_torque.A = mean_torque.A - offset.A*ones(1,size(mean_torque.A,2));
mean_torque.B = mean_torque.B - offset.B*ones(1,size(mean_torque.B,2));
mean_torque.C = mean_torque.C - offset.C*ones(1,size(mean_torque.C,2));
mean_torque.D = mean_torque.D - offset.D*ones(1, size(mean_torque.D,2));
mean_torque_ovrlpd.A = mean_torque_ovrlpd.A - offset_o.A*ones(1,...
                         size(mean_torque_ovrlpd.A,2));
mean_torque_ovrlpd.B = mean_torque_ovrlpd.B - offset_o.B*ones(1,...
                         size(mean_torque_ovrlpd.B,2));
mean_torque_ovrlpd.C = mean_torque_ovrlpd.C - offset_o.C*ones(1,...
                         size(mean_torque_ovrlpd.C,2));
mean_torque_ovrlpd.D = mean_torque_ovrlpd.D - offset_o.D*ones(1,...
                         size(mean_torque_ovrlpd.D,2));
%% SELEZIONE INGRESSI PER LE RETI NEURALI DA ADDESTRARE
\% ingressi per le reti addestrate con la tecnica delle finestre contigue
input_net_j2=catsamples(num2cell(mav_emg.A(1:3,:),1),num2cell(mav_emg.B(1:3,:),1),...
    num2cell(mav_emg.C(1:3,:),1), num2cell(mav_emg.D(1:3,:),1),'pad');
input_net_j4=catsamples(num2cell(mav_emg.A(4:5,:),1),num2cell(mav_emg.B(4:5,:),1),...
    num2cell(mav_emg.C(4:5,:),1),
                                       num2cell(mav_emg.D(4:5,:),1) ,'pad');
\% ingressi per le reti addestrate con la tecnica delle finestre sovrapposte
input_net_j2_o = catsamples(num2cell(mav_emg_ovrlpd.A([1 2 3],:),1),
  num2cell(mav_emg_ovrlpd.B([1 2 3],:),1), num2cell(mav_emg_ovrlpd.C([1 2 3],:),1),...
  num2cell(mav_emg_ovrlpd.D([1 2 3],:),1),'pad');
input_net_j4_o = catsamples(num2cell(mav_emg_ovrlpd.A(4:5,:),1),...
  num2cell(mav_emg_ovrlpd.B(4:5,:),1),num2cell(mav_emg_ovrlpd.C(4:5,:),1),...
  num2cell(mav_emg_ovrlpd.D(4:5,:),1)
                                           ,'pad');
```

```
%% SELEZIONE USCITE PER LE RETI NEURALI DA ADDESTRARE
% uscite per le reti addestrate con la tecnica delle finestre contigue
```

```
target_net_j2=catsamples(num2cell(mean_torque.A(1,:),1), num2cell(mean_torque.B(1,:),1),...
num2cell(mean_torque.C(1,:),1), num2cell(mean_torque.D(1,:),1),'pad');
target_net_j4=catsamples(num2cell(mean_torque.A(2,:),1), num2cell(mean_torque.B(2,:),1),...
num2cell(mean_torque.C(2,:),1), num2cell(mean_torque.D(2,:),1),'pad');
% uscite per le reti addestrate con la tecnica delle finestre sovrapposte
target_net_j2_o=catsamples(num2cell(mean_torque_ovrlpd.A(1,:),1),...
num2cell(mean_torque_ovrlpd.B(1,:),1),num2cell(mean_torque_ovrlpd.C(1,:),1),...
```

num2cell(mean_torque_ovrlpd.D(1,:),1), 'pad');

```
target_net_j4_o=catsamples(num2cell(mean_torque_ovrlpd.A(2,:),1),...
      num2cell(mean_torque_ovrlpd.B(2,:),1),num2cell(mean_torque_ovrlpd.C(2,:),1),...
      num2cell(mean_torque_ovrlpd.D(2,:),1),'pad');
%% ADDESTRAMENTO RETE SPALLA CON FINESTRE CONTIGUE
n_delays = 3;
                   % numero di delay del primo strato hidden
n_u = 1;
                    % numero di uscite della rete
F_j2 = n_delays*3; % numero di features in ingresso (3 muscoli * n_delays ritardi)
n_w_j2 = 700;
                    % numero pesi
n_h = 2;
                   % numero strati hidden
%numero di neuroni per strato
N = double(round(max(roots([(n_h-1) ((F_j2 + n_u)+(n_h)) (-n_w_j2+n_u)]))));
neuroni = N*ones(1,n_h); % neuroni nascosti della rete
clear temp_net
temp_net = timedelaynet(1:n_delays, neuroni);
temp_net.divideParam.trainRatio = 0.7;
temp_net.divideParam.valRatio = 0.3;
temp_net.divideParam.testRatio = 0;
temp_net.trainParam.max_fail = 8;
temp_net.trainParam.goal = 0.001;
temp_net.trainParam.showCommandLine = 1;
temp_net.trainParam.showWindow = 0;
[Xs,Xi,Ai,Ts] = preparets(temp_net,input_net_j2,target_net_j2);
temp_train = 100000;
for ntrain = 1:40
    temp_net = init(temp_net);
    [temp_net,temp_stats_net,~,~,~,~] = train(temp_net,Xs,Ts,Xi,Ai);
    if (temp_stats_net.best_vperf < temp_train) % bestnet</pre>
        temp_train = temp_stats_net.best_vperf;
        net_spalla = temp_net;
        stats_net_spalla = temp_stats_net;
    end
end
mse_A.spalla = plot_prestazioni(net_spalla, mav_emg.A([1 2 3],:), mean_torque.A(1,:));
mse_B.spalla = plot_prestazioni(net_spalla, mav_emg.B([1 2 3],:), mean_torque.B(1,:));
mse_C.spalla = plot_prestazioni(net_spalla, mav_emg.C([1 2 3],:), mean_torque.C(1,:));
mse_D.spalla = plot_prestazioni(net_spalla, mav_emg.D([1 2 3],:), mean_torque.D(1,:));
%% ADDESTRAMENTO RETE GOMITO CON FINESTRE CONTIGUE
                 % numero di delay del primo strato hidden
n_delays = 3;
n_u = 1;
                    % numero di uscite della rete
F_j4 = n_delays*2; % numero di features in ingresso (3 muscoli * n_delays ritardi)
n_w_j4 = 200;
                    % numero pesi
n_h = 2;
                   % numero strati hidden
%numero di neuroni per strato
N = double(round(max(roots([(n_h-1) ((F_j4 + n_u)+(n_h)) (-n_w_j4+n_u)]))));
neuroni = N*ones(1,n_h); % neuroni nascosti della rete
clear temp_net
temp_net = timedelaynet(1:n_delays, neuroni);
temp_net.divideParam.trainRatio = 0.7;
temp net.divideParam.valRatio = 0.3;
temp_net.divideParam.testRatio = 0;
temp_net.trainParam.max_fail = 8;
temp_net.trainParam.goal = 0.001;
temp_net.trainParam.showCommandLine = 1;
temp_net.trainParam.showWindow = 0;
[Xs,Xi,Ai,Ts] = preparets(temp_net,input_net_j4,target_net_j4);
```

 $temp_train = 100000;$

```
for ntrain = 1:40
    temp_net = init(temp_net);
    [temp_net,temp_stats_net,~,~,~,~] = train(temp_net,Xs,Ts,Xi,Ai);
    if (temp_stats_net.best_vperf < temp_train) % bestnet</pre>
        temp_train = temp_stats_net.best_vperf;
        net_gomito = temp_net;
         stats_net_gomito = temp_stats_net;
    end
end
mse_A.gomito = plot_prestazioni(net_gomito, mav_emg.A([4 5],:), mean_torque.A(2,:));
mse_B.gomito = plot_prestazioni(net_gomito, mav_emg.B([4 5],:), mean_torque.B(2,:));
mse_C.gomito = plot_prestazioni(net_gomito, mav_emg.C([4 5],:), mean_torque.C(2,:));
mse_D.gomito = plot_prestazioni(net_gomito, mav_emg.D([4 5],:), mean_torque.D(2,:));
%% ADDESTRAMENTO RETE SPALLA CON FINESTRE SOVRAPPOSTE
n_{delays} = 5;
                     % numero di delay del primo strato hidden
                     % numero di uscite della rete
n_u = 1;
F_j2 = n_delays*3; % numero di features in ingresso (3 muscoli * n_delays ritardi)
                      % numero pesi
n_w_j2 = 1400;
n_h = 2;
                     % numero strati hidden
%numero di neuroni per strato
N = double(round(max(roots([(n_h-1) ((F_j2 + n_u)+(n_h)) (-n_w_j2+n_u)]))));
neuroni = N*ones(1,n_h); % neuroni nascosti della rete
clear temp_net
temp_net = timedelaynet(1:n_delays, neuroni);
temp_net.divideParam.trainRatio = 0.7;
temp_net.divideParam.valRatio = 0.3;
temp_net.divideParam.testRatio = 0;
temp_net.trainParam.max_fail = 8;
temp_net.trainParam.goal = 0.001;
temp_net.trainParam.showCommandLine = 1;
temp_net.trainParam.showWindow = 0;
[Xs,Xi,Ai,Ts] = preparets(temp_net,input_net_j2_o,target_net_j2_o);
temp_train = 100000;
for ntrain = 1:40
    temp_net = init(temp_net);
    [temp_net,temp_stats_net,~,~,~] = train(temp_net,Xs,Ts,Xi,Ai);
if (temp_stats_net.best_vperf < temp_train) % bestnet</pre>
        temp_train = temp_stats_net.best_vperf;
        net_spalla_o = temp_net;
        stats_net_spalla_o = temp_stats_net;
    end
end
mse_A.spalla_o=plot_prestazioni(net_spalla_o,mav_emg_ovrlpd.A(1:3,:),...
    mean_torque_ovrlpd.A(1,:),0);
mse_B.spalla_o=plot_prestazioni(net_spalla_o,mav_emg_ovrlpd.B(1:3,:),...
    mean_torque_ovrlpd.B(1,:),0);
mse_C.spalla_o=plot_prestazioni(net_spalla_o,mav_emg_ovrlpd.C(1:3,:),...
    mean_torque_ovrlpd.C(1,:),0);
mse_D.spalla_o=plot_prestazioni(net_spalla_o,mav_emg_ovrlpd.D(1:3,:),...
    mean_torque_ovrlpd.D(1,:),0);
%% ADDESTRAMENTO RETE GOMITO CON FINESTRE SOVRAPPOSTE
                     \% numero di delay del primo strato hidden
n delavs = 5:
n_u = 1;
                     % numero di uscite della rete
F_{j4} = n_{delays*2};
                     % numero di features in ingresso (2 muscoli * n_delays ritardi)
n_w_j4 = 600;
                     % numero pesi
n_h = 2;
                     % numero strati hidden
%numero di neuroni per strato
```

```
N = double(round(max(roots([(n_h-1) ((F_j4 + n_u)+(n_h)) (-n_w_j4+n_u)]))));
neuroni = N*ones(1,n_h); % neuroni nascosti della rete
clear temp_net
temp_net = timedelaynet(1:n_delays, neuroni);
temp_net.divideParam.trainRatio = 0.7;
temp_net.divideParam.valRatio = 0.3;
temp_net.divideParam.testRatio = 0;
temp_net.trainParam.max_fail = 8;
temp_net.trainParam.goal = 0.001;
temp_net.trainParam.showCommandLine = 1;
temp_net.trainParam.showWindow = 0;
[Xs,Xi,Ai,Ts] = preparets(temp_net,input_net_j4_o,target_net_j4_o);
temp_train = 100000;
for ntrain = 1:40
    temp_net = init(temp_net);
[temp_net,temp_stats_net,~,~,~] = train(temp_net,Xs,Ts,Xi,Ai);
    if (temp_stats_net.best_vperf < temp_train) % bestnet</pre>
        temp_train = temp_stats_net.best_vperf;
        net_gomito_o
                       = temp_net;
        stats_net_gomito_o = temp_stats_net;
    end
end
mse_A.gomito_o = plot_prestazioni(net_gomito_o, mav_emg_ovrlpd.A([4 5],:),...
      mean_torque_ovrlpd.A(2,:),0);
mse_B.gomito_o = plot_prestazioni(net_gomito_o, mav_emg_ovrlpd.B([4 5],:),...
      mean_torque_ovrlpd.B(2,:),0);
mse_C.gomito_o = plot_prestazioni(net_gomito_o, mav_emg_ovrlpd.C([4 5],:),...
      mean_torque_ovrlpd.C(2,:),0);
mse_D.gomito_o = plot_prestazioni(net_gomito_o, mav_emg_ovrlpd.D([4 5],:),...
      mean_torque_ovrlpd.D(2,:),0);
```

A.3.4 filtro_ecg.m

```
f = 256;
                      % frequenza di campionamento [Hz]
wl = 0.2;
                     % larghezza finestra (200 ms)
wd = 0.1;
                     \% distanza tra due finestre sovrapposte
ncw = floor(f*wl); % numero di campioni della finestra
NWL = ncw;
                      % larghezza finestra in samples
D = floor(f*wd); % numero di campioni di distanza tra due finestre sovrapposte
Of = 26; % ordine del filtro per l'ecg
[b,a] = butter(2,0.6,'low'); % filtro coppia al gomito
[Bhe Ahe] = butter(4,20/(256/2),'high');
[Bh Ah] = butter(2,25/(256/2),'low');
t_in = find(dati.Data(:,23),1,'first');
t_end = find(dati.Data(:,23),1,'last');
ECG = filter(Bh,Ah,dati.Data(t_in:t_end,4));
EMG = dati.Data(t_in:t_end,1);
H = estimateh(EMG,ECG,f,Of);
save filtro_ecg_acquisizione_DATA.mat H
clear t_in t_end ECG EMG
```

A.3.5 plot_prestazioni.m

Calcolo della prestazione sul dataset di test della rete net e visualizzazione dei grafici temporali.

function MSE = plot_prestazioni(net,input,target,flag)
% calcola l'errore quadratico medio calcolato per la rete NET, per l'uscita

```
% predetta rispetto all'uscita TARGET
%
% se flag è pari ad 1 visualizza i grafici di autocorrelazione dell'errore
% e di correlazione tra ingresso ed errore
[Xs,Xi,Ai,Ts] = preparets(net,num2cell(input,1),num2cell(target,1));
Y = net(num2cell(input,1),Xi,Ai);
E = gsubtract(num2cell(target,1),Y); % calcola l'errore rispetto al target dato
if(flag == 1)
    figure; ploterrcorr(E)
    figure; plotinerrcorr(num2cell(input,1),E)
end
figure;plot(cell2mat(Y)),hold,plot(target,'red'),legend('T','\tau')
MSE = mse(E);
```

A.4 Interfaccia grafica per l'acquisizione

A.4.1 EMG_Lexos_GUI_isometric_light

Codice dell'interfaccia grafica per l'acquisizione dei punti.

function [sys,x0,str,ts,simStateCompliance]=EMG_Lexos_GUI_isometric_light(t,x,u,flag)

```
persistent u_old;
persistent rip_old;
persistent P_old;
if isempty(u_old)
    u old = 0;
    rip_old = 0;
    P_old = 0;
end
%
                     2
                             3
                                          4
                                                   5
                                                        6
                                                               7
                                                                             8
            1
letters = {'ECG', 'riposo', 'avanti', 'indietro', 'su', 'giù', 'stop', 'invaliducommand'};
      = \{ 'A', 'B', 'C', 'D' \};
Pee
%gui parameter runmode: %standard = 0, no feedback = 1, no commands (free run) = 2
runmode = 0;
global fig;
switch flag,
  % % % % % % % % % % % % % % % % % %
  % Initialization %
  % % % % % % % % % % % % % % % % %
  case 0,
    [sys,x0,str,ts,simStateCompliance]=mdlInitializeSizes;
    fig = figure (113);
    pos = get(0, 'MonitorPositions');
    if size(pos,1) > 1 % se c'è più di un monitor, sposta la gui nel secondo
       pos=[pos(1,3)+1 1 800 600];% con due monitor, sposta la finestra sul secondo monitor
       set(fig,'Position',pos,'MenuBar','none');
    else % se c'è un solo monitor, posiziona la gui nel primo monitor
        pos = pos(1,:);
        set(fig,'Visible','off');
        set(fig,'Position',pos,'MenuBar','none');
        movegui(fig,'center');
    end
```

```
axis([-100 100 -100 100]);
  axis('off');
  hold on;
  handles.rectangle=rectangle('Position', [-100 -100 200 200], 'FaceColor', 'k',...
         'Visible','on');
  handles.line1 = line([0 0], [50 -50],'LineWidth',2,'Visible','on','Color','w');
handles.line2 = line([-50 50], [0 0],'LineWidth',2,'Visible','on','Color','w');
handles.text1 = text(0,30,letters{1},'FontSize',100,'Color','w','Visible','off');
  handles.text2 = text(70,70,'0','FontSize',80,'Color','b','Visible','off');
  handles.text3 = text(70,70,Pee{1},'FontSize',100,'Color','b','Visible','off');
  set(gca,'UserData',handles);
  set(fig,'Visible','on');
%%%%%%%%%%%%%%%%%%
% Derivatives %
%%%%%%%%%%%%%%%%%%%
case 1.
  sys=mdlDerivatives(t,x,u);
% % % % % % % % % %
% Update %
%%%%%%%%%%%
case 2,
  sys=mdlUpdate(t,x,u);
  handles=get(gca, 'UserData');
  if(u(1) ~= u_old) %managing task texts
       if (u(1)>0 && runmode ~= 2)
         set(handles.text1,'Visible','off');
         set(handles.text2,'Visible','off');
         app=get(handles.text1,'Visible');
         if strcmp(app,'off')
           switch(u(1))
              case 1 % CALCOLO FILTRO ECG
                 handles.text1=text(0,0,letters{1},'FontSize',150,...
                 'HorizontalAlignment', 'center', 'Color', 'r', 'Visible', 'off');
                 set(handles.text1,'Visible','on');
              case 2 % REGISTRAZIONE FASE REST
                handles.text1=text(0,0,letters{2},'FontSize',150,...
                 'HorizontalAlignment', 'center', 'Color', 'w', 'Visible', 'off');
                 set(handles.text1,'Visible','on');
              case 3 % REGISTAZIONE MOVIMENTI IN AVANTI
                 handles.text1=text(0,30,letters{3},'FontSize',150,...
                    'HorizontalAlignment','center','Color','b','Visible','off');
                 set(handles.text1,'Visible','on');
              case 4 % REGISTAZIONE MOVIMENTI INDIETRO
                 handles.text1=text(0,30,letters{4},'FontSize',150,...
                   'HorizontalAlignment', 'center', 'Color', 'g', 'Visible', 'off');
                 set(handles.text1, 'Visible', 'on');
              case 5 % REGISTAZIONE MOVIMENTI IN SU
                 handles.text1=text(0,30,letters{5},'FontSize',150,...
                   'HorizontalAlignment', 'center', 'Color', 'b', 'Visible', 'off');
                 set(handles.text1,'Visible','on');
              case 6 % REGISTAZIONE MOVIMENTI IN GIU'
                 handles.text1=text(0,30,letters{6},'FontSize',150,...
                   'HorizontalAlignment', 'center', 'Color', 'g', 'Visible', 'off');
                 set(handles.text1,'Visible','on');
              case 7 % STOP RECORDING
                 handles.text1=text(0,0,letters{7},'FontSize',150,...
                 'HorizontalAlignment', 'center', 'Color', 'r', 'Visible', 'off');
```
```
set(handles.text1,'Visible','on');
            case 8 % INVALID COMMAND
              handles.text1=text(0,0,letters{8},'FontSize',80,...
              'HorizontalAlignment', 'center', 'Color', 'y', 'Visible', 'off');
               set(handles.text1,'Visible','on');
            otherwise
              handles.text1=text(0,0,letters{8},'FontSize',100,...
              'HorizontalAlignment', 'center', 'Color', 'y', 'Visible', 'off');
               set(handles.text1,'Visible','on');
         end
      end
    else
         set(handles.text1,'Visible','off');
    end
end
if(u(1) ~= u_old) %enabling/disabling fixation cross
    if(u(1)==0) % || (u(1)>=3 && u(1)<=6))
         set(handles.line1, 'Visible','on');
set(handles.line2, 'Visible','on');
    else
         set(handles.line1, 'Visible','off');
set(handles.line2, 'Visible','off');
    end
end
if(u(2) ~= rip_old) %managing text of repetitions
  if (u(2) >= 0)
   set(handles.text2,'Visible','off');
   app2=get(handles.text2,'Visible');
   if strcmp(app2,'off')
    if (u(1)>2 \&\& u(1)<7) %mostra il numero delle ripetizioni fatte
      handles.text2=text(100,100,num2str(u(2)),'FontSize',100,
       'HorizontalAlignment','right','VerticalAlignment','top','Color','b',...
       'Visible', 'off');
      set(handles.text2, 'Visible', 'on');
    end
   else
   set(handles.text2,'Visible','off');
  end
 end
end
if(u(3) ~= P_old) %managing text of Pee
  if (u(3) >= 1)
     set(handles.text3,'Visible','off');
     handles.text3=text(70,100,Pee{u(3)},'FontSize',100,'HorizontalAlignment',...
'right','VerticalAlignment','top','Color','w','Visible','on');
  else
     set(handles.text3,'Visible','off');
  end
else
  if(u(1)<=1 || u(1)>=7)
    if u(3)>=1
       if strcmp(get(handles.text3,'Visible'),'off');
         handles.text3=text(70,100,Pee{u(3)},'FontSize',100,'HorizontalAlignment',...
           'right','VerticalAlignment','top','Color','w','Visible','on');
      end
    end
  end
```

```
drawnow;
    set(gca, 'UserData', handles);
    sys=[];
    u_old = u(1); % vecchio valore dello stato
   rip_old = u(2); % vecchio valore delle ripetizioni
P_old = u(3); % vecchio valore della posizione
 %%%%%%%%%%%%%
 % Outputs %
 %%%%%%%%%%%%%
 case 3,
   sys=mdlOutputs(t,x,u);
 % GetTimeOfNextVarHit %
 case 4,
   sys=mdlGetTimeOfNextVarHit(t,x,u);
 %%%%%%%%%%%%%%%
 % Terminate %
 %%%%%%%%%%%%%%%
 case 9.
    sys=mdlTerminate(t,x,u);
 % % % % % % % % % % % % % % % % % % % %
  % Unexpected flags %
 otherwise
   DAStudio.error('Simulink:blocks:unhandledFlag', num2str(flag));
end
% end sfuntmpl
% mdlInitializeSizes
% Return the sizes, initial conditions, and sample times for the S-function.
function [sys,x0,str,ts,simStateCompliance]=mdlInitializeSizes
\% call simsizes for a sizes structure, fill it in and convert it to a
% sizes array.
\% Note that in this example, the values are hard coded. This is not a
\% recommended practice as the characteristics of the block are typically
\% defined by the S-function parameters.
sizes = simsizes;
sizes.NumContStates = 0;
sizes.NumDiscStates = 0;
sizes.NumOutputs = 0;
                    = 3;
sizes.NumInputs
sizes.DirFeedthrough = 0;
sizes.NumSampleTimes = 1;
                             % at least one sample time is needed
sys = simsizes(sizes);
\% initialize the initial conditions
x0 = [];
% str is always an empty matrix
str = [];
\% initialize the array of sample times
ts = [-1 \ 0];
```

end

```
\% Specify the block simStateCompliance. The allowed values are:
%
     'UnknownSimState', < The default setting; warn and assume DefaultSimState
     'DefaultSimState', < Same sim state as a built-in block
%
     'HasNoSimState', < No sim state
'DisallowSimState' < Error out when saving or restoring the model sim state
%
%
simStateCompliance = 'UnknownSimState';
% end mdlInitializeSizes
% mdlDerivatives
\% Return the derivatives for the continuous states.
function sys=mdlDerivatives(t,x,u)
sys = [];
% end mdlDerivatives
% mdlUpdate
% Handle discrete state updates, sample time hits, and major time step
% requirements.
function sys=mdlUpdate(t,x,u)
sys = [];
% end mdlUpdate
% mdlOutputs
% Return the block outputs.
function sys=mdlOutputs(t,x,u)
svs = []:
% end mdlOutputs
% mdlGetTimeOfNextVarHit
% Return the time of the next hit for this block. Note that the result is
\% absolute time. Note that this function is only used when you specify a
\% variable discrete-time sample time [-2 0] in the sample time array in
% mdlInitializeSizes.
function sys=mdlGetTimeOfNextVarHit(t,x,u)
% sampleTime = 1;
                    % Example, set the next hit to be one second later.
sampleTime = 1/256;
                       % Example, set the next hit to be one second later.
sys = t + sampleTime;
% end mdlGetTimeOfNextVarHit
% mdlTerminate
% Perform any end of simulation tasks.
function sys=mdlTerminate(t,x,u)
sys = [];
% end mdlTerminate
```