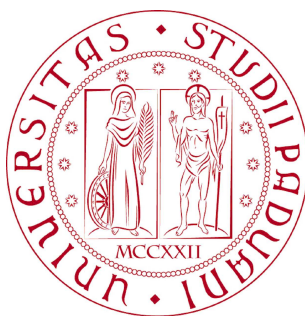




UNIVERSITÀ DEGLI STUDI DI PADOVA

DIPARTIMENTO DI TECNICA E GESTIONE DEI SISTEMI
INDUSTRIALI

DIPARTIMENTO DI INGEGNERIA INDUSTRIALE

CORSO DI LAUREA MAGISTRALE IN INGEGNERIA
MECCATRONICA

CONTROLLO ADATTATIVO DI UN ROBOT A CAVI PER LA RIABILITAZIONE DELL'ARTO SUPERIORE

RELATORE: CH.MO PROF. ING. GIOVANNI BOSCHETTI

LAUREANDO: PIETRO MATTIOLO

MATRICOLA: 2027754

ANNO ACCADEMICO 2022-2023

Al Pietro del passato: alla fine ci sei arrivato, hai visto?

“ If we hold on, there’s a victory over the Sun. ”

BIFFY CLYRO

Indice

Sommario	XI
Introduzione	XIII
1 La robotica riabilitativa	1
1.1 L'ictus cerebrale	1
1.2 Trattamento e riabilitazione	2
1.3 Robotica riabilitativa	4
1.3.1 Esoscheletro	4
1.3.2 End-effector	8
1.3.3 Robot a cavi	9
1.4 Efficacia della robotica riabilitativa	11
1.5 Obiettivi del lavoro	12
2 MariBot e MariBot 2.0	15
2.1 MariBot	15
2.1.1 Apparato meccanico	15
2.1.2 Funzionamento	17
2.1.3 Controllo	18
2.2 MariBot 2.0	20
2.2.1 Apparato meccanico	20
2.2.2 Controllo	25
3 Sistema elettrico e azionamenti	27
3.1 Quadri elettrici	27
3.1.1 Quadro di alimentazione	27

3.1.2	Quadro di potenza e controllo	35
3.2	Sensori e altri elementi	39
3.3	Azionamenti	40
3.4	Dimensionamento del quadro elettrico	43
3.4.1	Dimensionamento elettrico dei motori dei link	43
3.4.2	Dimensionamento elettrico dei motori dei cavi	44
3.4.3	Scelta degli alimentatori	45
3.5	Cablaggio	46
3.5.1	Collegamenti di alimentazione e finecorsa	47
3.5.2	Collegamenti driver	48
3.5.3	Segnali encoder	51
4	Algoritmi di controllo e event manager	55
4.1	Logica di funzionamento complessivo	56
4.2	Macchina a stati	59
4.2.1	Segnali usati per la sincronizzazione	60
4.2.2	Flag, write e diagramma di flusso	63
4.3	Implementazione degli stati	69
4.3.1	Controllo in posizione	69
4.3.2	Controllo in coppia: tensionatori	70
4.3.3	Acquisizione	71
4.3.4	Pianificazione traiettoria	73
5	Controllo a basso livello	79
5.1	Controllore PID	79
5.2	Simulazioni cavi	81
5.2.1	Modello del sistema	81
5.2.2	Risultati simulativi cavi	83
5.3	Simulazioni giunti	88
5.3.1	Modello del sistema	88
5.3.2	Risultati simulativi giunti	92
5.4	Risultati sperimentali	94
5.4.1	Risultati sperimentali cavi	94

5.4.2	Risultati sperimentali giunti	98
6	Controllo ad alto livello	101
6.1	Controllo adattativo con forgetting factor	101
6.1.1	Radial Basis Functions	102
6.1.2	Vettore dei pesi stimati	104
6.1.3	Schema di controllo	107
6.2	Simulazione per controllo adattativo del singolo cavo	107
6.2.1	Schema simulativo di controllo	108
6.2.2	Risultati simulativi singolo cavo	111
6.3	Risultati sperimentali controllo adattativo	122
6.3.1	Risultati sperimentali singolo cavo	122
6.3.2	Risultati sperimentali tre cavi	130
6.3.3	Risultati sperimentali giunti	142
	Conclusioni	155
A	Implementazione pianificatore traiettoria	159
A.1	Implementazione pianificatore traiettoria	159
A.1.1	Ricerca delle posizioni da interpolare	159
A.1.2	Precalcolo degli istanti di passaggio	160
A.1.3	Generazione traiettoria di riferimento	161
	Bibliografia	165

Sommario

La robotica è, per definizione, quella branca dell'ingegneria che si occupa di creare dispositivi che siano in grado di replicare i movimenti degli esseri umani per sostituirsi ad essi e svolgere al loro posto lavori usuranti o nei quali sia richiesta elevata precisione, velocità o forza.

I robot impiegati nella riabilitazione, tuttavia, non devono semplicemente essere in grado di sostituire completamente l'azione dell'uomo, ma devono cooperare insieme ad esso per raggiungere l'obiettivo desiderato, che è spesso rappresentato da un definito movimento di terapia.

Questo aspetto rappresenta una sfida che sta vedendo, negli ultimi anni, lo sviluppo di diverse soluzioni. Tali soluzioni mirano a rendere il macchinario quanto più collaborativo possibile e capace di adattarsi in modo automatico alle diverse risposte che pazienti in condizioni differenti possono avere durante l'esercizio, lasciando dunque al terapeuta il compito della configurazione iniziale e della supervisione, dando la possibilità potenziale di gestire più pazienti nello stesso momento.

Nei laboratori del DII (Dipartimento di Ingegneria Industriale) dell'Università di Padova si sta sviluppando, partendo dai primi anni 2000, il *MariBot*, un robot per la riabilitazione dell'arto superiore per pazienti post-ictus in fase acuta e sub-acuta.

La presente tesi interesserà le migliorie apportate nella sua versione *MariBot 2.0*, concentrandosi soprattutto sugli aspetti riguardanti il controllo e il cablaggio.

Introduzione

La seguente tesi costituisce un tassello del mosaico che si sta componendo nel corso degli anni e che rappresenta la realizzazione complessiva del MariBot 2.0, braccio robotico per la riabilitazione dell'arto superiore per pazienti post-ictus in fase acuta e sub-acuta.

Ciò che andrà a trattare questo elaborato, nel dettaglio, sarà lo sviluppo della logica di controllo che gestisce il funzionamento del dispositivo aptico in tutte le sue fasi, prestando particolare attenzione alla costruzione di un controllore adattivo per i motori dei giunti robotici di tipo *assist-as-needed* con *forgetting factor* durante la terapia.

Questo particolare tipo di controllore consente al manipolatore di adattare il proprio output in funzione della performance del paziente: riducendola quando l'esercizio viene eseguito correttamente e aumentandola qualora questo non si verificasse. In questo modo si cerca di ottenere un robot capace di “premiare” gli sforzi del soggetto ma al contempo correggerlo nei casi in cui dovesse servire. L'implementazione di tutti gli schemi di controllo sarà fatta in ambiente *MATLAB* e *Simulink Real Time*.

Un ulteriore aspetto affrontato in questo lavoro è la gestione della componentistica elettronica e dei cablaggi che permettono il funzionamento del robot.

Di seguito si riporta una breve panoramica degli argomenti trattati nei diversi capitoli della tesi.

Il capitolo 1 riguarderà la descrizione dell'ictus e di come questo possa essere trattato attraverso processi riabilitativi, prestando particolare attenzione alle terapie assistite da robot.

Nel capitolo 2 sarà presentato il MariBot e le modifiche che hanno portato alla

nascita della sua evoluzione, MariBot 2.0.

Il capitolo 3 vedrà la descrizione dell'hardware impiegato nella realizzazione della parte elettronica del manipolatore e del cablaggio realizzato per garantirne il funzionamento.

Nel capitolo 4 verrà descritta l'implementazione della macchina a stati utilizzata per la gestione del controllo complessivo del manipolatore.

Il capitolo 5 riporterà un'analisi del comportamento dei cavi e dei giunti del robot in caso di controllo a basso livello (di tipo proporzionale-integrale-derivativo).

Infine nel capitolo 6 si presenterà la teoria e le prove riguardanti il controllo adattativo e la sua implementazione per la gestione dei cavi e dei giunti del robot.

Capitolo 1

La robotica riabilitativa

I danni neurologici provocati da ictus o traumi possono determinare un impatto più o meno grave sulla capacità di movimento nel soggetto interessato, risultando nella difficoltà di compiere anche normali azioni quotidiane.

Per cercare di migliorare e ristabilire le abilità motorie danneggiate si adottano delle tecniche di terapia riabilitativa che consistono nel far compiere al paziente una serie di movimenti in maniera ripetitiva.

In questa fase l'impiego di dispositivi aptici e robot risulta vantaggioso sia dal punto di vista del terapeuta, il quale vede notevolmente ridotto il proprio carico di lavoro, che del paziente. Recenti studi hanno infatti dimostrato che i migliori risultati si ottengono proprio da trattamenti terapeutici eseguiti con l'assistenza di apparati robotici.

In questo capitolo saranno illustrati alcuni aspetti medici inerenti all'ictus e si fornirà una descrizione dello stato dell'arte dei robot impiegati per la riabilitazione.

1.1 L'ictus cerebrale

Per ictus si intende una sindrome clinica determinata dall'occlusione (ictus ischemico) o rottura (ictus emorragico) di un'arteria cerebrale che porta alla comparsa di un deficit neurologico focale o, più raramente, globale.

Si conta che nel mondo sono circa 15 milioni le persone che vengono colpite da ictus ogni anno. Di queste, 5 milioni vanno incontro a morte e altri 5 milioni subiranno danni che porteranno loro a condizioni di disabilità permanente [1]. La gravità dei danni che l'ictus comporta sulla persona è strettamente connessa al tempo di intervento e ospedalizzazione. Il tempestivo riconoscimento dei sintomi e la conseguente erogazione di cure mediche adeguate, infatti, rappresentano la chiave per ridurre, quanto più possibile, le lesioni al cervello determinate da questo disturbo. Il tempo entro il quale l'infortunato dovrebbe ricevere assistenza medica è di circa tre ore.

Grazie al miglioramento e alla diffusione delle campagne di prevenzione, oltre che a un più attento controllo dei fattori di rischio, è stato possibile ridurre l'incidenza di questa patologia del 42% negli ultimi quarant'anni. I sopra citati fattori di rischio per ictus vengono distinti tra non modificabili e modificabili.

Tra quelli non modificabili rientrano l'età, il sesso, l'etnia, il corredo genetico e l'ereditarietà; esempi di fattori modificabili sono invece: ipertensione arteriosa, fibrillazione o cardiopatia atriale, diabete, elevati livelli di colesterolo, uso di droghe, alcol e fumo, diete poco sane e inattività fisica.

1.2 Trattamento e riabilitazione

Le fasi successive all'ospedalizzazione di un soggetto colpito da ictus rappresentano la vera sfida nel suo processo di guarigione.

L'ictus può determinare lesioni cerebrali di diversa entità e con ripercussioni più o meno evidenti: disflasia, afasia, difficoltà motorie o completa paralisi di uno o più arti. Risulta dunque evidente che il cammino riabilitativo non sarà univoco ma dovrà essere progettato ad-hoc per il singolo individuo da personale sanitario specializzato.

L'obiettivo della riabilitazione è quindi quello di sfruttare i sistemi neurali del soggetto rimasti integri, stimolandoli nell'apprendimento di competenze e abilità che possano migliorare quanto possibile il controllo della persona, in modo da ridurre e limitare quella percezione di malessere psicologico che le limitazioni im-

poste dall'evento ictale comporta.

È possibile suddividere il processo di riabilitazione post-ictus in tre fasi [2]:

- **Fase acuta (“della prevenzione”):**

dura circa due settimane e viene tipicamente eseguita nelle Stroke Unit dei reparti di Neurologia, specializzate nel trattamento di pazienti in questa fase della malattia.

- **Fase sub-acuta (“del recupero”):**

ha durata variabile ma comunque compresa tra uno e sei mesi, viene eseguita all'interno dei reparti o centri di riabilitazione da fisioterapisti o logopedisti.

- **Fase cronica (“degli esiti”):**

ulteriori 6-12 mesi di terapia che conducono alla fine del percorso riabilitativo.

Entrando nello specifico, le linee guida del Ministero della Sanità [3] suddividono le attività riabilitative in due famiglie:

- **attività di riabilitazione intensiva:**

della durata di almeno tre ore al giorno con elevato impegno assistenziale e professionale.

- **attività di riabilitazione estensiva o intermedia:**

richiedono un impegno riabilitativo di durata compresa fra una e tre ore giornaliere con un forte impegno assistenziale.

Recenti studi hanno dimostrato che elevata intensità e assiduità nello svolgimento degli esercizi terapeutici garantiscono migliori risultati nel recupero. [4]

Dovendo quindi prestare assistenza praticamente costantemente per tutta la durata della seduta è evidente che l'ausilio di robot che riescano a cooperare con il fisioterapista nello svolgimento degli esercizi alleggerisca notevolmente il lavoro di quest'ultimo. *Neurorehabilitation and Neural Repair* [5] riporta, infatti, che un punto di forza dell'impiego di dispositivi aptici in campo riabilitativo sia proprio la possibilità di poter eseguire più facilmente una terapia ad alta intensità, grazie anche al fatto che la quantità di movimenti generati da questo tipo di terapia è di gran lunga superiore rispetto a quello ottenibile da altre forme di riabilitazione.

1.3 Robotica riabilitativa

Fin dai primi anni 2000 l'interesse nei confronti della robotica funzionale riabilitativa è in costante aumento.

La causa del recente sviluppo di questa branca della medicina si può cercare nel generale miglioramento delle condizioni di vita e della maggior probabilità di sopravvivenza a malattie, cui consegue una crescita della popolazione anziana e del numero di persone con disabilità.

Dai dati riportati dal OMS, nel 2018 erano circa un miliardo gli individui che vivevano con una qualche sorta di handicap, tra cui quelli causati da danni neurologici con ripercussioni sulle funzioni motorie [6]. Sono proprio questi i soggetti, più o meno anziani, per i quali i robot per la riabilitazione risultano un mezzo fondamentale verso la guarigione. Poiché si prevede che entro il 2050 la popolazione mondiale raggiungerà i 9,7 miliardi di persone [7], il numero di individui destinato a trattamenti terapeutici di questo tipo sarà anch'esso destinato a crescere.

Questi fattori hanno determinato l'espansione del mercato di dispositivi aptici e fornito una spinta nella ricerca di nuove tecnologie e sistemi che consentano di ottenere trattamenti sempre più efficienti ed efficaci.

Risulta pertanto utile classificare i diversi tipi di manipolatori per la riabilitazione. Tale classificazione è possibile sia dal punto di vista della logica funzionale, la quale prevede la distinzione in robot basati su esoscheletro o su end-effector che in base all'architettura costruttiva, la quale permette di individuare nel dettaglio la famiglia dei robot a cavi.

1.3.1 Esoscheletro

Per “esoscheletro” si intende un sistema indossabile dal paziente capace di seguirne i movimenti, i quali sono pianificati direttamente nello spazio dei giunti dell'essere umano.

Per far sì che l'apparato esoscheletrico riesca a replicare al meglio i movimenti imposti dalla terapia, è necessario che esista corrispondenza tra i giunti dell'essere

umano e quelli del robot.

Questo significa, ad esempio, che una corretta flessione del gomito del paziente è ottenibile solo se l'esoscheletro è composto da un accoppiamento rotoidale con asse quanto più coincidente possibile con quello dell'avambraccio. Minore l'allineamento tra questi assi e maggiore sarà il contributo di forze traslazionali durante la terapia, non desiderate e spesso dolorose per il soggetto sotto cura.

Evidentemente l'eterogeneità nelle caratteristiche anatomiche della popolazione richiede sforzi mirati a rendere quanto più versatile possibile la struttura del dispositivo robotico, in modo che questo possa essere utilizzato per diversi pazienti, previa configurazione da parte del terapista.

Un esempio di robot basato su esoscheletro per la riabilitazione dell'arto superiore è il MEDARM (*Motorized Exoskeleton Device for Advanced Rehabilitation of Motor function*), del 2007 [Figura 1.1] [8]. Per garantire la condizione sull'allineamento degli assi umani e robotici esposta in precedenza, questo robot sfrutta una struttura con assi di rotazione supplementari capaci di modificare la posizione del giunto di spalla e gomito dell'apparato esoscheletrico, adattandolo da paziente a paziente. Come si evince anche dalla figura riportata, il sistema ha un ingombro notevole ed è pertanto da intendersi come una soluzione che richiede installazione fissa.

Questo aspetto rappresenta una debolezza del MEDARM, in quanto spesso i robot per la riabilitazione devono essere spostati tra le diverse camere dei reparti ospedalieri e adattati alla posizione del letto o sedia del paziente; risulta pertanto necessario che essi siano quanto possibile maneggevoli e facilmente manovrabili. Rimanendo nell'ambito della riabilitazione dell'arto superiore, nel 2013 è stato sviluppato il NEUROExos (*NEURObotics Elbow eXOSkeleton*) [Figura 1.2] [9]. Esso risulta composto di 13 giunti passivi i quali, oltre che correggere il posizionamento degli assi di rotazione come il MEDARM, tengono anche conto del lassismo dell'articolazione del gomito. Durante il movimento di flessione dell'avambraccio, infatti, l'asse anatomico non rimane fisso e con esso dovrà muoversi anche quello dell'esoscheletro.

Per quanto concerne la riabilitazione dell'arto inferiore con esoscheletro, sempre nel 2007 nasce ALEX (*ActiveLeg EXoskeleton*) [Figura 1.3] [10], il quale imple-

menta un approccio innovativo al controllo delle forze tra robot e paziente che è quello dell'assist-as-needed.

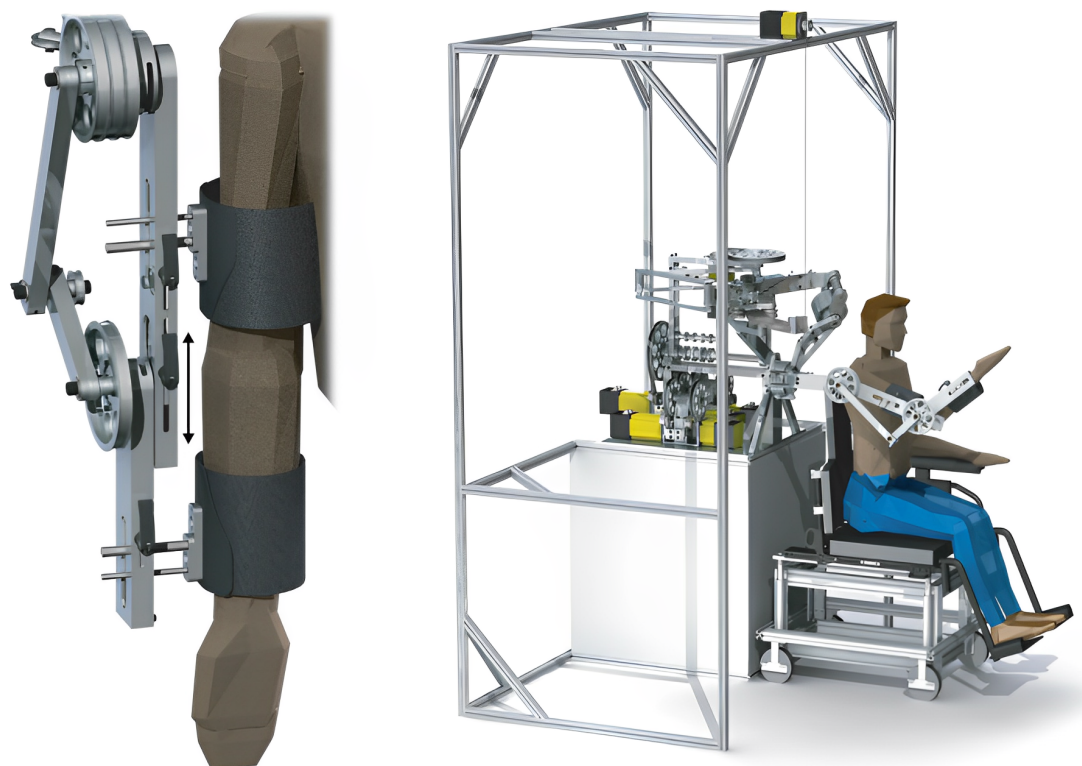


Figura 1.1: Esoscheletro MEDARM per la riabilitazione dell'arto superiore

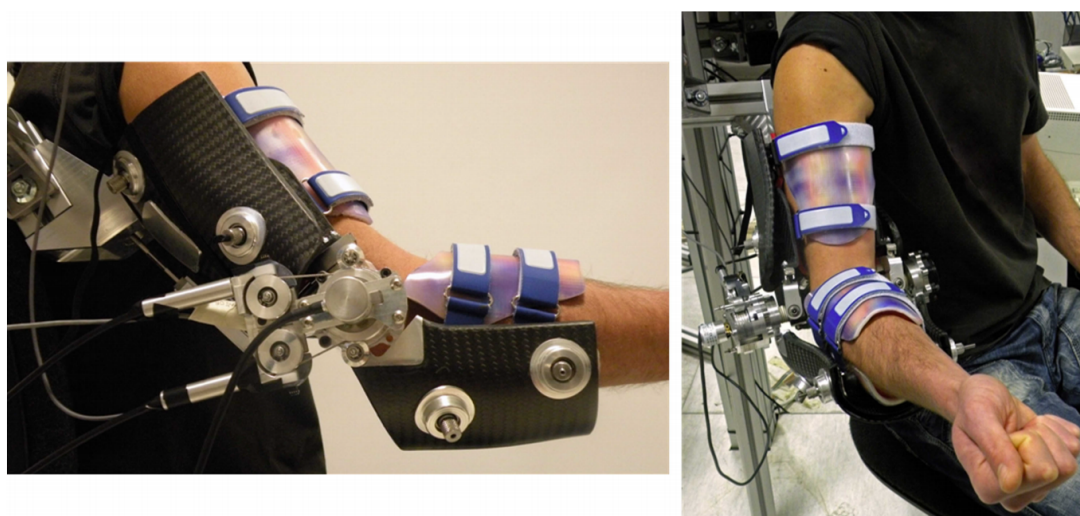


Figura 1.2: Esoscheletro NUROExos per la riabilitazione dell'arto superiore



Figura 1.3: Esoscheletro ALEX per la riabilitazione dell'arto inferiore

1.3.2 End-effector

I sistemi aptici basati su end-effector interagiscono con il paziente agganciandosi ad esso solo alla parte terminale del segmento corporeo da riabilitare. Il collegamento avviene in un unico punto di contatto (“effettore” o “ortesi”) e pertanto non esiste quella correlazione tra giunti umani e robotici che invece caratterizzava gli apparati esoscheletrici.

Questo aspetto fa sì che la pianificazione dell’esercizio terapeutico debba essere eseguita nello spazio dei giunti del robot oppure nello spazio cartesiano, eliminando la necessità di allineamento tra assi di rotazione umani e robotici. È inoltre da evidenziare come l’impiego di ortesi, tipicamente meno avvolgenti e invasive di un esoscheletro, si traduce da un lato in maggior confort per il paziente, ma dall’altro a un minor controllo dei movimenti dei tratti non vincolati all’effettore.

Tra i robot riabilitativi con end-effector è necessario citare il MIT-MANUS [Figura 1.4] [11], [12], primo dispositivo per la riabilitazione assistita sviluppato dal M.I.T. nel 1997 e tutt’oggi ancora in commercio. Il suo impiego è rivolto mirato alla terapia di pazienti con danni neurologici conseguenti all’ictus che hanno determinato ripercussioni sulle capacità motorie dell’arto superiore.

In questo manipolatore l’end-effector è rappresentato da una sorta di joystick che permette al paziente di disegnare su uno spazio virtuale (visualizzato tramite display) il movimento terapeutico richiesto.

La meccanica del MANUS è quella di un manipolatore SCARA a cinque link che permette la movimentazione sui due gradi di libertà planari ai quali, grazie a un meccanismo differenziale, è possibile aggiungere ulteriori tre gradi di libertà per il polso del paziente. Un secondo esempio di robot con end-effector è il NeReBot (*NEuroREhabilitation roBOT*) [Figura 1.5] [13], sviluppato nel Laboratorio di Robotica del DIMEG (Dipartimento di Innovazione Meccanica E Gestionale) dell’Università degli Studi di Padova nel 2005. Come il MIT-MANUS anche il NeReBot è destinato alla riabilitazione dell’arto superiore. In questo caso però l’effettore, un’ortesi solidale con l’intero avambraccio del paziente, esegue l’esercizio attraverso l’avvolgimento o svolgimento di tre cavi connessi ad altrettanti link



Figura 1.4: Robot MIT-MANUS per la riabilitazione dell'arto superiore

mobili, la cui orientazione è definita dal terapeuta prima dell'inizio dell'esercizio.

1.3.3 Robot a cavi

I manipolatori descritti in precedenza possono anche essere distinti in base all'architettura realizzativa che li caratterizza. Per fare alcuni esempi, il MIT-MANUS rientra nella categoria dei dispositivi SCARA; MEDARM, NEUROExos e NeReBot appartengono invece alla schiera dei robot a cavi.

Quest'ultima famiglia necessita di un particolare approfondimento, soprattutto perché anche il MariBot, fulcro di questa tesi, rientra nella suddetta categoria.

In questi dispositivi l'arto del paziente è movimentato attraverso dei cavi avvolti su pulegge, tipicamente azionate da motori elettrici. L'impiego di cavi comporta alcuni vantaggi, che verranno descritti di seguito.

Il primo vantaggio è la maggior leggerezza delle trasmissioni, la quale si ripercuote in minori inerzie e dunque minore sforzo motore richiesto. Un ulteriore aspetto positivo è l'elevato rapporto tra carico e massa, con conseguente aumento del



Figura 1.5: Robot NeReBot per la neuroriabilitazione dell'arto superiore

marginale di sicurezza durante l'utilizzo. I robot a cavi sono inoltre molto adatti ai trattamenti riabilitativi in quanto da un lato riescono a garantire il controllo del movimento desiderato senza essere eccessivamente invasivi o costringenti; dall'altro l'aspetto estetico meno industriale ha un impatto positivo verso il paziente, il quale si rende più disponibile e tollerante nei confronti della macchina. È inoltre da evidenziare come i robot di questo tipo siano tipicamente caratterizzati da minori costi ma comunque buone prestazioni.

Un esempio di robot a cavi, oltre a quelli sopra citati, è il Sophia-3 e Sophia-4 (*String Operated Planar Haptic Interface for upper-Arm rehabilitation*) [14][Figura 1.6], sviluppati dal DIMEG nel 2008.

Questi sono interfacce aptiche per la riabilitazione post-ictus dell'arto superiore ad end-effector con un concetto simile al MIT-MANUS ma in cui la componente meccanica dello SCARA e del meccanismo differenziale è sostituita da 4 (Sophia-4) o 3 (Sophia-3) cavi. Il tensionamento di questi cavi riesce a trasmettere al paziente sensazioni di forza resistente, che dovrà essere vinta durante la terapia.

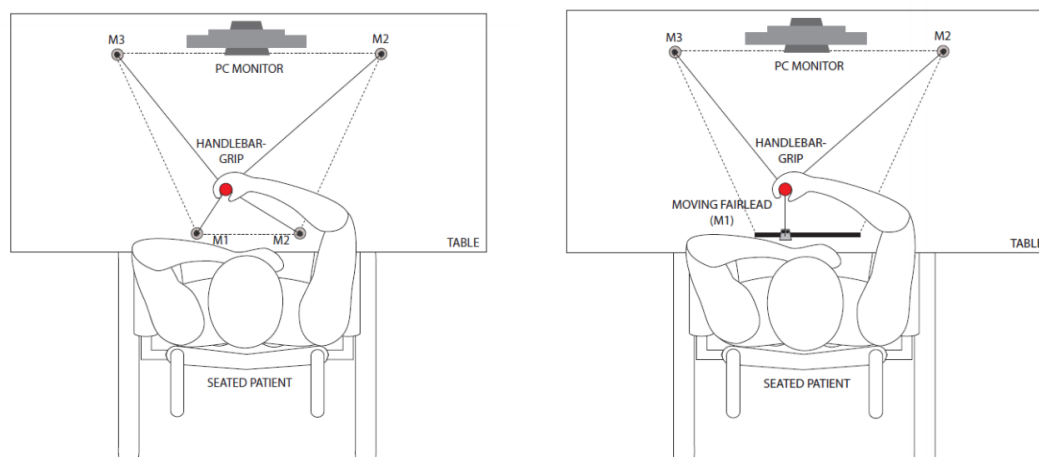


Figura 1.6: Sophia-4 (a sinistra) e Sophia-3 (a destra)

1.4 Efficacia della robotica riabilitativa

Recenti studi hanno dimostrato, come visto in precedenza, che il trattamento assistito da robot di pazienti in fase acuta e sub-acuta risulta essere più efficace rispetto alle terapie convenzionali, soprattutto se il piano terapeutico prevede sedute ad elevata intensità fin dalle prime fasi successive all'evento ictale.

Questo può essere dimostrato analizzando i risultati di una serie di studi [15],[16] svolti sul NeReBot in aggiunta e in parziale sostituzione della terapia classica. Si è infatti constatato come i soggetti che avessero ricevuto fin dalla fase acuta una stimolazione sensomotoria, usufruendo delle terapie di NeReBot, mostrino una notevole riduzione della disabilità motoria e un netto miglioramento delle attività funzionali.

Nei grafici di Figura 1.7 è illustrato l'andamento degli indicatori FMA, MRC e FIM relativi al gruppo di controllo (CG) e al gruppo sperimentale (EG) in esame.

- **FMA (Fugl-Meyer Assessment):**

fornisce un punteggio da 0 a 66 in merito alla disabilità motoria dell'arto. L'assegnazione del punteggio risulta ripartita in 42/66 punti relativi alla parte prossimale del braccio (spalla, gomito e avambraccio) e 24/66 punti relativi alla parte distale (polso e mano).

- **MRC (Medical Research Council):**

esprime il livello di forza muscolare ed assegna un valore compreso tra 0 e 6, spaziando dalla completa impossibilità di movimento fino alla capacità di movimento anche contro una resistenza massima applicata. Il grafico illustrato si fa riferimento al deltoide durante il movimento di abduzione del braccio.

- **FIM (Functional Independence Measure):**

valutazione dell'abilità del paziente nel compiere 18 attività della vita quotidiana di cui 13 sono di tipo motorio e 5 di tipo cognitivo. Ognuna di queste attività può essere valutata con un punteggio da 1 a 7, dove gli estremi stanno rispettivamente per la completa dipendenza dagli altri e per la totale autosufficienza.

Come si evince dai risultati riportati l'efficacia dell'ausilio del NeReBot fin dalla fase immediatamente successiva all'ictus (1 settimana) è indiscutibile. È tuttavia necessario aggiungere che i miglioramenti delle condizioni del paziente siano sempre strettamente dipendenti dal terapeuta, la cui azione risulta decisiva ai fini del recupero ottimale soprattutto per quanto concerne la configurazione iniziale del dispositivo.

Il MariBot, e conseguentemente la sua evoluzione MariBot 2.0, mirano quindi a porre rimedio agli aspetti negativi del loro predecessore nello stesso campo di impiego.

1.5 Obiettivi del lavoro

L'obiettivo centrale di questa tesi è l'implementazione di un nuovo schema di controllo complessivo per MariBot 2.0, prestando particolare attenzione allo sviluppo di un controllore adattativo per i link mirato a migliorare l'interazione paziente-robot.

Per far ciò si è reso necessario modernizzare la parte elettronica del manipolatore, pertanto si fornirà una descrizione dei dispositivi utilizzati e cablaggi realizzati.

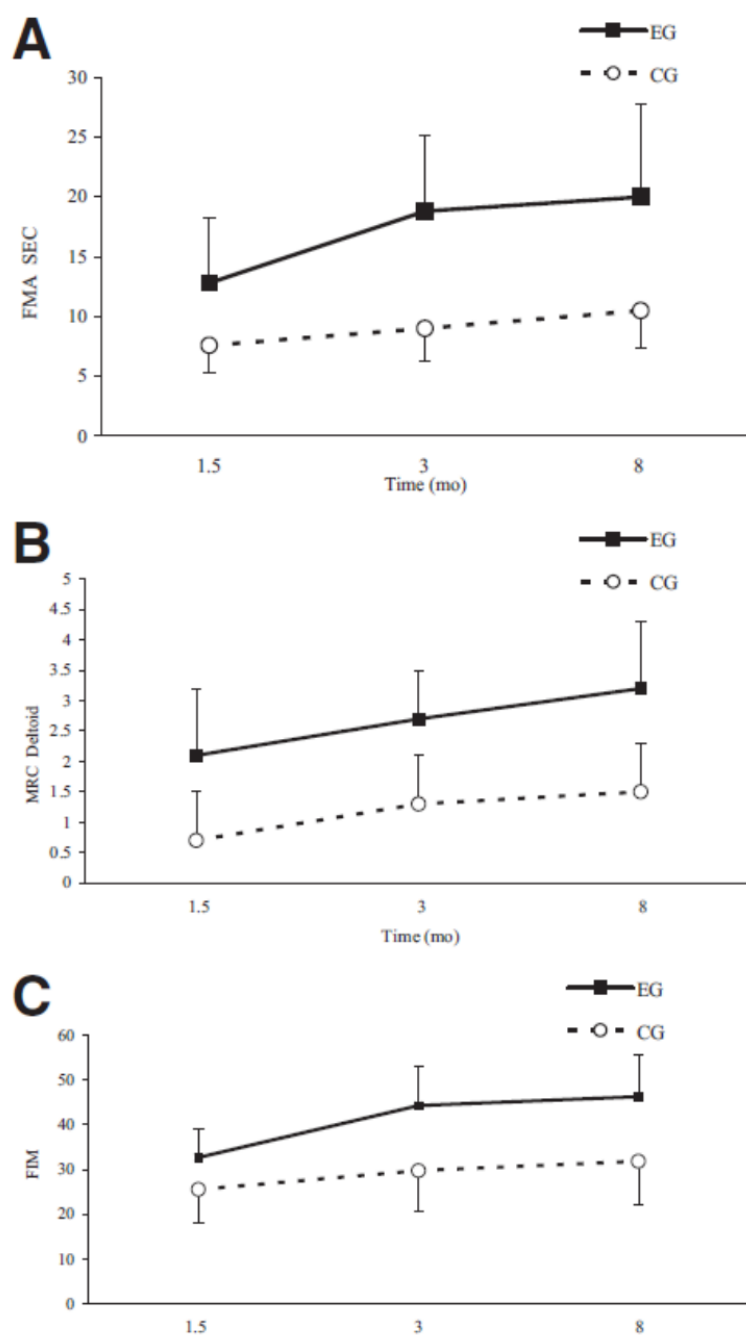


Figura 1.7: Confronto del recupero motorio a lungo termine (8 mesi): A) FMA, B) MRC, C) FMI

Capitolo 2

MariBot e MariBot 2.0

In questa sezione saranno presentati la struttura meccanica, il controllo e il funzionamento del MariBot. Si metteranno inoltre in evidenza le modifiche che hanno condotto allo sviluppo di una sua versione evoluta, il MariBot 2.0.

Ciò a cui si presterà maggior attenzione sarà nel dettaglio la parte inerente il controllo del manipolatore.

2.1 MariBot

2.1.1 Apparato meccanico

Il MariBot prende il nome dal sollevapazienti Arjo Marisa, che costituisce la struttura portante del robot [Figura 2.1] [17]. Questa struttura, capace di sorreggere masse elevate (capacità di carico 195 kg), resta comunque facilmente movimentabile grazie alle ruote di cui essa è dotata e inoltre adattabile nelle procedure di posizionamento grazie alle sue gambe regolabili, che possono essere poste sotto ai piedi di un letto ospedaliero, garantendo quindi la possibilità di terapia anche a pazienti allettati.

Alla colonna verticale del sollevatore è connesso il braccio meccanico del manipolatore, adattabile in altezza per mezzo di una guida lineare azionata da un motore elettrico dedicato. Tale braccio è composto di tre membri seriali (link): LINK 0, fisso, permette di lasciare lo spazio necessario per il posizionamento del paziente

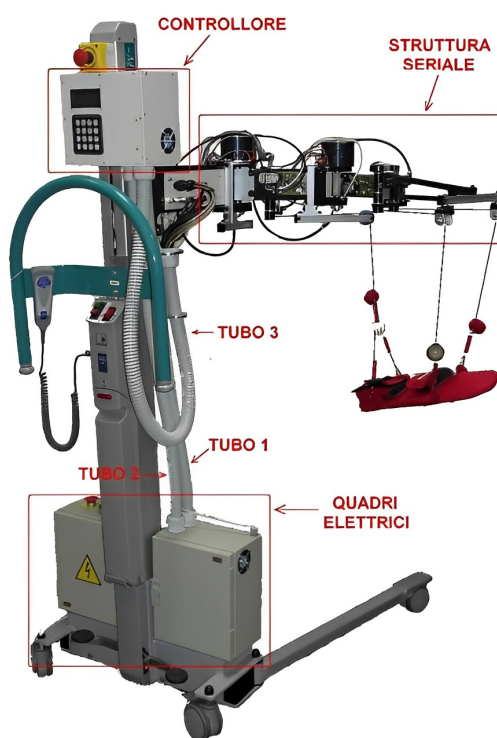
al di sotto dello stesso; LINK 1 e LINK 2, di fatto costituiscono un classico robot SCARA. Il corretto posizionamento del paziente prevede che la sua spalla si trovi esattamente sotto il giunto J1 tra LINK 0 e LINK 1 (il punto di connessione tra LINK 1 e LINK 2 è denominato, invece, J2).

Il movimento dei membri mobili è ottenuto sfruttando due motori brushless in presa diretta indipendenti, montati direttamente sul giunto di competenza [Figura 2.2].

Dal braccio seriale dipartono, per mezzo di un sistema di pulegge e passacavi,



(a)



(b)

Figura 2.1: a) Sollevatore Marisa; b) MariBot

tre cavi che permettono il sostegno dell'arto del paziente per mezzo dell'ortesi. La connessione tra ortesi e cavi avviene attraverso tre magneti. Ciascuno dei tre cavi è gestito attraverso un motoriduttore controllato in modo indipendente dagli altri, due posizionati sul LINK 2 (gestiscono il sollevamento e rotazione del polso)

e uno sul LINK 1 (permette l'alzata del gomito).

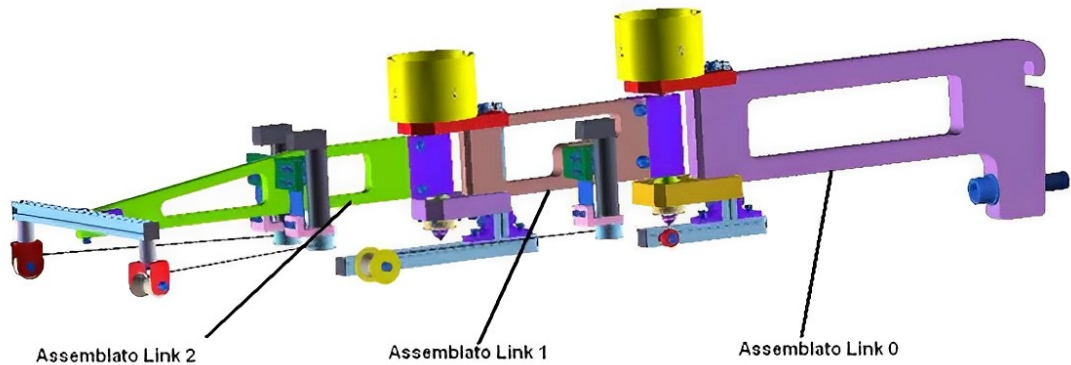


Figura 2.2: Disegno CAD del braccio meccanico (progetto esecutivo)

L'ortesi, che rappresenta l'end-effector del manipolatore, ha dunque il fondamentale compito di fissare una corrispondenza tra i movimenti del robot e quelli del braccio del paziente e viceversa. Essa rappresenta l'unico oggetto del robot con il quale il soggetto dovrà entrare in contatto e pertanto è necessario che risulti poco invadente e confortevole durante tutte le fasi della terapia. A tal fine essa presenta una serie di agganci a velcro e di diversi fori per l'aggancio dei cavi, con il fine di adattarsi al meglio alle caratteristiche anatomiche dei diversi pazienti cui è destinata la riabilitazione. Nel caso in esame l'ortesi utilizzata è denominata "Baita" [Figura 2.3].

2.1.2 Funzionamento

Il funzionamento del MariBot vede la suddivisione del processo di riabilitazione in due fasi principali: fase di acquisizione e fase di terapia.

- **Fase di acquisizione:**

può anche essere definita come "apprendimento", in quanto essa definisce la fase in cui il terapeuta, sostenendo il braccio del paziente, gli fa percorrere il movimento terapeutico desiderato e, mano a mano che si percorre la traiettoria



Figura 2.3: a) Ortesi “Baita”; b) Dettaglio dell’ortesi

voluta, memorizza una serie di punti attraverso la pressione di un pulsante sull’interfaccia di controllo.

Da questi punti il controllore sarà in grado di generare le leggi di moto che rappresenteranno poi i riferimenti da seguire in fase di terapia.

In questo stadio il controllo deve prevedere entrambi i motori dei LINK inattivi, in modo che sia possibile manovrare agevolmente il braccio robotico; i cavi dovranno essere invece mantenuti in controllo di coppia per poter consentire di modificare la lunghezza dei cavi senza che essi si svolgano troppo, garantendo dunque una forza di tensionamento costante.

- **Fase di terapia:**

tutti i motori sono ora controllati in posizione e permettono l’esecuzione del movimento riabilitativo stabilito nella fase precedente.

2.1.3 Controllo

La gestione del controllo di un dispositivo aptico complesso come il MariBot richiede di affrontare e risolvere molteplici problematiche.

Per prima cosa è necessario definire l’hardware che si andrà ad utilizzare, il quale dovrà essere necessariamente in grado di garantire requisiti di *Real Time*. Questa caratteristica rappresenta la capacità del controllore di fornire risposte adeguate

entro un certo tempo massimo (*deadline*), tipicamente dell'ordine dei millisecondi.

Nella prima versione del manipolatore il controllo è affidato a una serie di PC/104, computer integrati il cui nome deriva dai 104 pin del connettore esterno che permette di rendere queste schede modulari, ciascuna adibita a una diversa funzione ma ugualmente interconnesse tra loro.

Un altro aspetto inerente la componentistica è la scelta dei mezzi trasmissivi, cioè della tipologia di cablaggio impiegato. Dopo successive evoluzioni [18], la versione finale del MariBot consente di ridurre la quantità di cavi che portano i segnali di comando e di alimentazione dai quadri elettrici posizionati sulla base del manipolatore verso gli attuatori posti invece sul braccio, anche grazie alla scelta di adottare motori con encoder integrati. Per quanto concerne i driver utilizzati, cioè quei componenti elettronici il cui scopo è adattare l'alimentazione proveniente dalla rete al corretto impiego desiderato, si utilizzano i motori CC Maxon Re-26 (posteriore) e Re-25 (frontali) per i cavi, e il modello K089075 della Baysidemotion per i link.

Per completare il controllo occorre infine specificare l'ambiente software utilizzato e la logica che lo governa.

La programmazione, eseguita in linguaggio C/C++, è stata implementata su sistema operativo VxWorks, capace di soddisfare i vincoli imposti dal *Real Time*. Volendo analizzare il problema del controllo del MariBot (e conseguentemente anche di MariBot 2.0) nel suo funzionamento completo, si possono individuare due sfide principali. La prima è rappresentata dalla gestione della sincronizzazione tra le diverse fasi del programma, che richiede la commutazione tra diversi schemi di controllo dei motori al verificarsi di determinate condizioni. La seconda problematica è poi la gestione del robot in fase di terapia, cioè durante l'esecuzione del movimento terapeutico appreso in fase di acquisizione.

Il problema della sincronizzazione può essere considerato come la necessità di una corretta implementazione, più o meno articolata, di una macchina a stati finiti

¹, dunque esso non va ad impattare sull'efficacia del trattamento terapeutico, in quanto questa dipendono solamente dal controllo definito in fase di esercizio, che rappresenta il secondo aspetto descritto pocanzi.

Per MariBot si è optato come approccio al controllo della movimentazione riabilitativa un controllo in posizione basato su controllore PID (Proporzionale Integrale Derivativo).

Come sarà ampiamente descritto più avanti in questa tesi, un controllo di questo tipo riesce effettivamente a garantire ottimi inseguimenti in posizione e velocità, rispettando dunque il percorso desiderato con una buona precisione, tuttavia lo fa senza richiedere lo sforzo del paziente che, se vogliamo 'subisce' la terapia.

2.2 MariBot 2.0

MariBot 2.0 si presenta come un'evoluzione del suo predecessore, con il quale condivide la struttura meccanica di base composta da Marisa, braccio seriale a due link e tre cavi e la logica generale di funzionamento, articolata nelle fasi di acquisizione e di terapia vera e propria.

Il nuovo manipolatore si promette in particolare di migliorare aspetti inerenti proprio la struttura meccanica e il controllo del braccio oltre che, dove possibile, modernizzare l'apparato elettronico con soluzioni più recenti e all'avanguardia

2.2.1 Apparato meccanico

Tra le modifiche che sono state introdotte nella versione 2.0 del robot è possibile individuarne alcune particolarmente importanti le quali hanno permesso, nel complesso, una miglior gestione della dinamica del manipolatore:

¹Rappresentazioni di sistemi dinamici che passano da una modalità di funzionamento (stato) all'altra

- Spostamento del motore del LINK2 a telaio e creazione di un quadrilatero articolato per la trasmissione del moto [Figura 2.4 FOTO e 2.5 CAD].



Figura 2.4: MariBot 2.0

L'obiettivo di tale upgrade risiede nel vantaggio ottenuto in termini di coppia richiesta ai motori dei link. In lavori di tesi precedenti ([19]) sono infatti stati confrontati dal punto di vista simulativo i contributi di coppia motrice

richiesta durante un tipico esercizio di terapia come la rotazione dell'arto attorno ad un asse fisso verticale tale da determinare un movimento risultante si sviluppa interamente sul piano, senza richiedere l'intervento dei cavi.

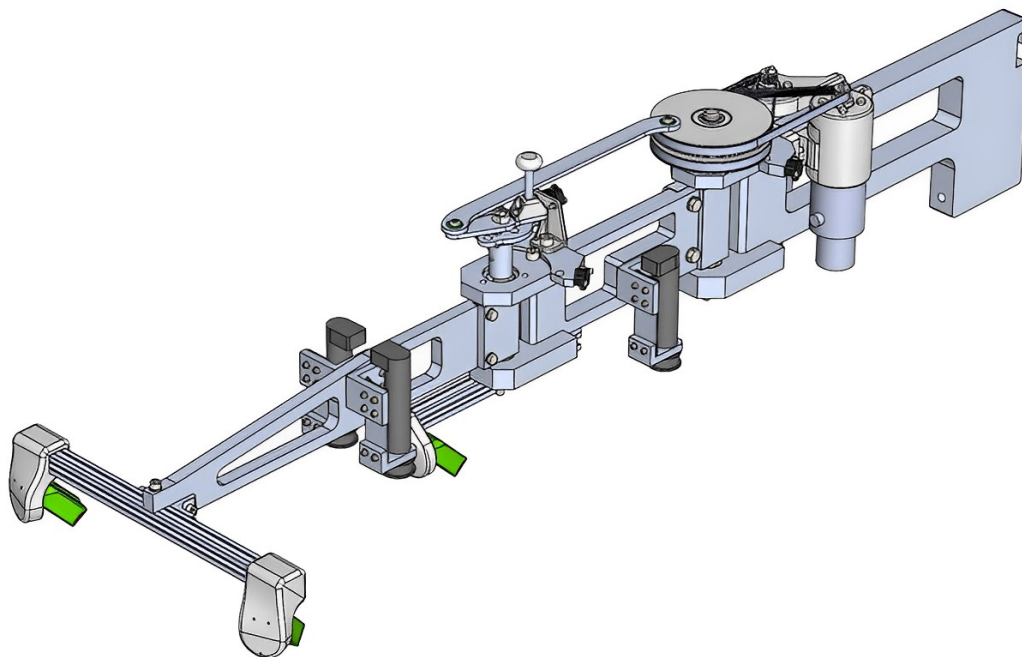


Figura 2.5: Disegno CAD del braccio meccanico (progetto esecutivo)

- Introduzione di rapporto di riduzione per i motori dei link.

La decisione di impiegare un riduttore di giri anche per l'attuazione dei link (accompagnandosi al sistema di riduzione ai cavi presente già da MariBot) è dovuta ai noti vantaggi che esso comporta durante l'utilizzo. Il primo di tali vantaggi risiede nella possibilità di adattamento al campo operativo, intendendo in questo senso la riduzione delle velocità e l'aumento delle coppie erogabili al carico, caratteristiche ben più che utili nell'ambito delle macchine per la movimentazione di carichi come possono essere per esempio i robot per la riabilitazione. Un ulteriore aspetto positivo è la linearizzazione della dinamica dettata dal fatto che eventuali variazioni nell'inerzia vista a lato carico vengono pesate per il rapporto di trasmissione del riduttore che, essendo una quantità minore di uno, le rende meno influenti rispetto ad un accoppiamento diretto tra componente motrice e load.

In Figura 2.7 è riportato il sistema di riduzione introdotto in MariBot 2.0,

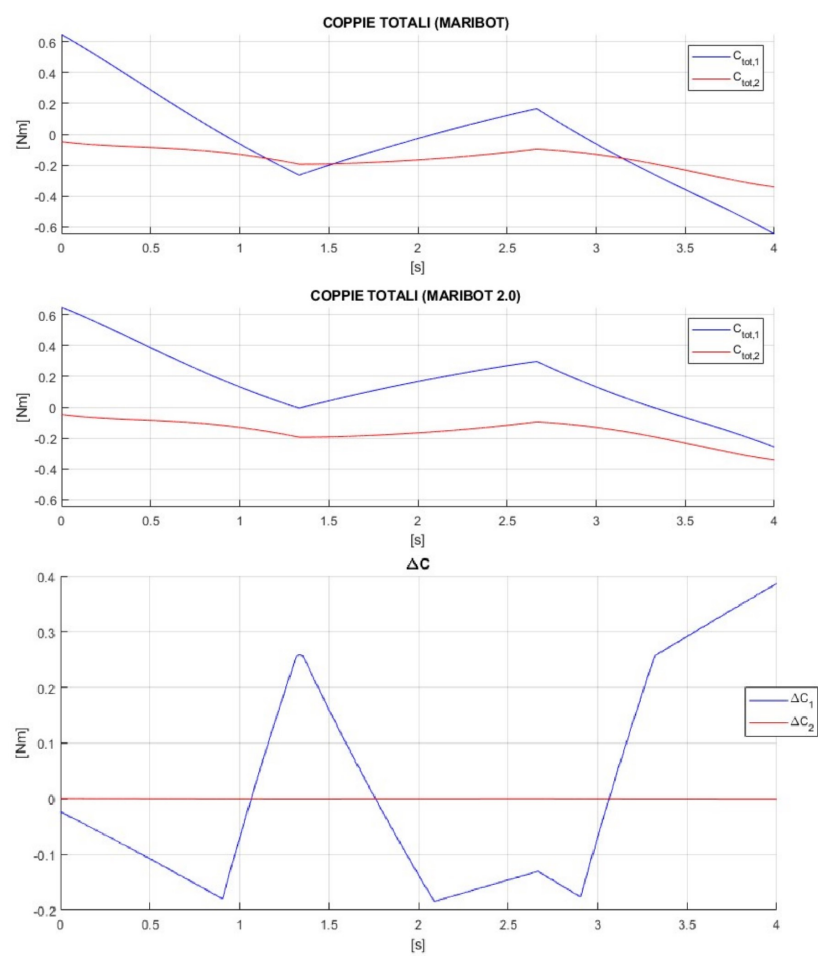


Figura 2.6: Differenza percentuale sulle coppie totali richieste ai giunti per MariBot e MariBot 2.0

uguale per entrambi i motori dei link e costituito da un accoppiamento tra cinghie dentate e pulegge.

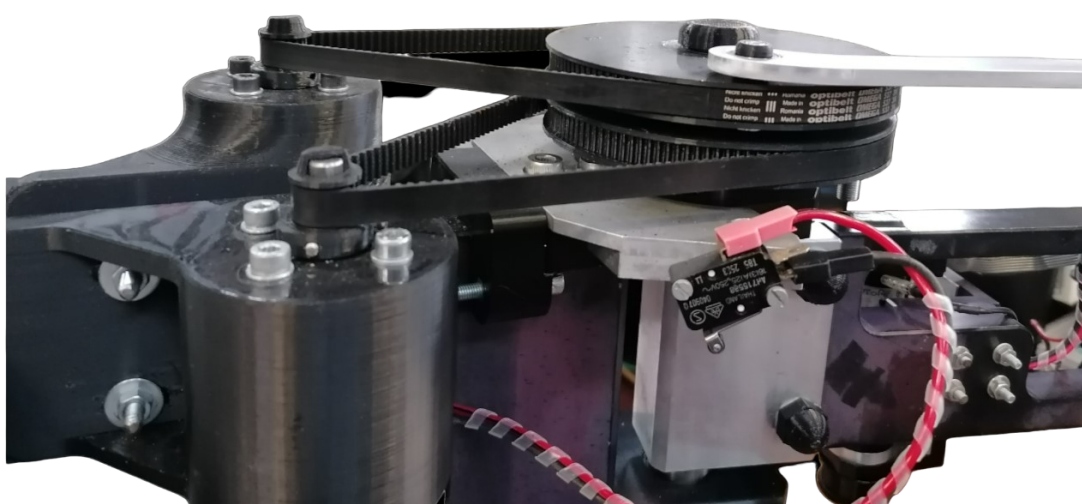


Figura 2.7: Rapporto di trasmissione per i link

- Meccanismo per la commutazione rapida da configurazione a braccio destro o sinistro.

Col fine di rendere MariBot 2.0 quanto più versatile possibile e adatto quindi alla riabilitazione di pazienti con necessità diverse, è stato necessario studiare un metodo che consentisse la possibilità di modificare la configurazione del braccio seriale in funzione dell'arto del paziente da sottoporre a terapia, se destro o sinistro.

Per ottenere questo obiettivo si rende però necessario andare a modificare la struttura a quadrilatero appena introdotta ogni qualvolta si decida di commutare da una configurazione all'altra. Evidentemente la modifica completa dell'intera struttura risulterebbe troppo laboriosa, complessa e decisamente scomoda. L'idea era quindi cercare di ottenere un sistema che consentisse una modifica rapida e semplice, effettuabile nel giro di pochi secondi dallo stesso terapista in fase di inizializzazione e settaggio iniziale.

Questo obiettivo è stato centrato con il dispositivo di sgancio a molla ri-

portato in Figura 2.8.



Figura 2.8: Meccanismo di sgancio a molla per cambio configurazione

2.2.2 Controllo

Il controllo di MariBot2.0 rappresenta il fulcro di questo lavoro di tesi e pertanto verrà trattato in maniera approfondita in tutti i suoi diversi aspetti nei capitoli che seguiranno. Risulta ad ogni modo importante evidenziare e fin d'ora i punti chiave che lo rendono diverso dal suo predecessore.

Partendo dall'aspetto hardware², MariBot2.0 presenta migliorie soprattutto sul lato del controllore: dai PC/104 si passa infatti a più moderni moduli Beckhoff di lettura e scrittura. Questi permettono un minore ingombro grazie alla possibilità di unire i diversi dispositivi ottenendo blocchi della dimensione adeguata all'impiego desiderato e inoltre la possibilità di essere impiegati in ambiente *Real Time* secondo protocollo EtherCAT, tecnologia per la comunicazione deterministica prettamente mirata all'impiego nel mondo industriale.

I software utilizzati per lo sviluppo e la gestione del controllore sono *MATLAB* e *Simulink Real-Time*, impiegati sia a livello simulativo che sperimentale. In essi è stato possibile realizzare la gestione della sincronizzazione delle diverse fasi attraverso la creazione di una macchina a stati (*Event manager*)³ e del controllore adattativo per la terapia.

²Descrizione dettagliata nel Capitolo 3

³Descrizione dettagliata nel Capitolo 5

Quest'ultimo rappresenta un punto fondamentale nello sviluppo del MariBot2.0 in quanto la tecnologia medica riabilitativa sta sempre più evolvendosi nella direzione di macchinari collaborativi, capaci di assistere il paziente nel giusto modo e nella giusta quantità, al contrario di manipolatori che forniscono un'assistenza passiva, in cui l'interazione uomo-macchina è simile a quella esistente tra un sollevatore e un carico, dunque nei quali il contributo umano e la cooperazione col robot sono ridotti al minimo.

La robotica riabilitativa, quindi, sta sempre più cercando di ottenere dispositivi aptici che siano accondiscendenti ("*compliant*") nei confronti della persona e capaci di rispondere alle sue precise esigenze durante l'esercizio, modificando l'apporto di forza erogato nell'assistenza in modo che vi sia sempre una componente data dallo sforzo umano. La strada da percorrere per perseguire questi obiettivi è quella di creare logiche di controllo basate su approcci adattativi.

Diversi studi infatti dimostrano come l'ausilio di robot che sfruttano questo tipo di controllori in pazienti con almeno una minima capacità motoria residua, porti da un lato a migliori risultati dal punto di vista clinico [20], con un tasso di recupero più elevato rispetto alla terapia classica. e dall'altro a un maggior interesse e propensione dell'individuo allo svolgimento dell'esercizio: la possibilità di modificare attivamente la risposta del dispositivo rende il paziente più propositivo nello svolgimento del movimento desiderato[21].

Nel caso di MariBot2.0 l'implementazione di un controllo adattivo è già stata apportata, in una prima versione, al controllo dei cavi di sostegno dell'ortesi. Questo elaborato si propone di migliorare e approfondire tale controllore e al contempo di estendere la stessa filosofia anche alla gestione dei motori dei link, in modo da ottenere un manipolatore interamente governato, almeno nella fase più importante del trattamento terapeutico, cioè l'esecuzione del percorso desiderato in maniera ciclica e intensiva, da una logica di controllo adattativo.

Capitolo 3

Sistema elettrico e azionamenti

Nel seguente capitolo si intende portare una descrizione della parte elettronica che garantisce il funzionamento di MariBot 2.0. Saranno quindi descritti i singoli componenti utilizzati e ne verrà presentato il relativo cablaggio. Per fornire una descrizione più efficace verranno esposti singolarmente gli aspetti inerenti a: quadri elettrici di alimentazione e di controllo, sensoristica e altri elementi, azionamenti e cablaggio. Si riporta quindi il criterio utilizzato per il dimensionamento che ha portato alla scelta dei vari componenti descritti.

3.1 Quadri elettrici

Per quanto riguarda la disposizione dei diversi elementi elettronici nel manipolatore, questi sono localizzati principalmente nei due quadri posti alla base del Marisa, anche se alcuni trovano la loro locazione in altri punti del braccio seriale. I numeri in giallo fanno riferimento all'elenco dei componenti riportato di seguito.

3.1.1 Quadro di alimentazione

La componentistica di alimentazione è quella utilizzata per adattare la tensione proveniente dalla rete all'impiego nel robot, garantendo comunque che questa venga erogata solamente quando tutte le condizioni di sicurezza sono rispettate.

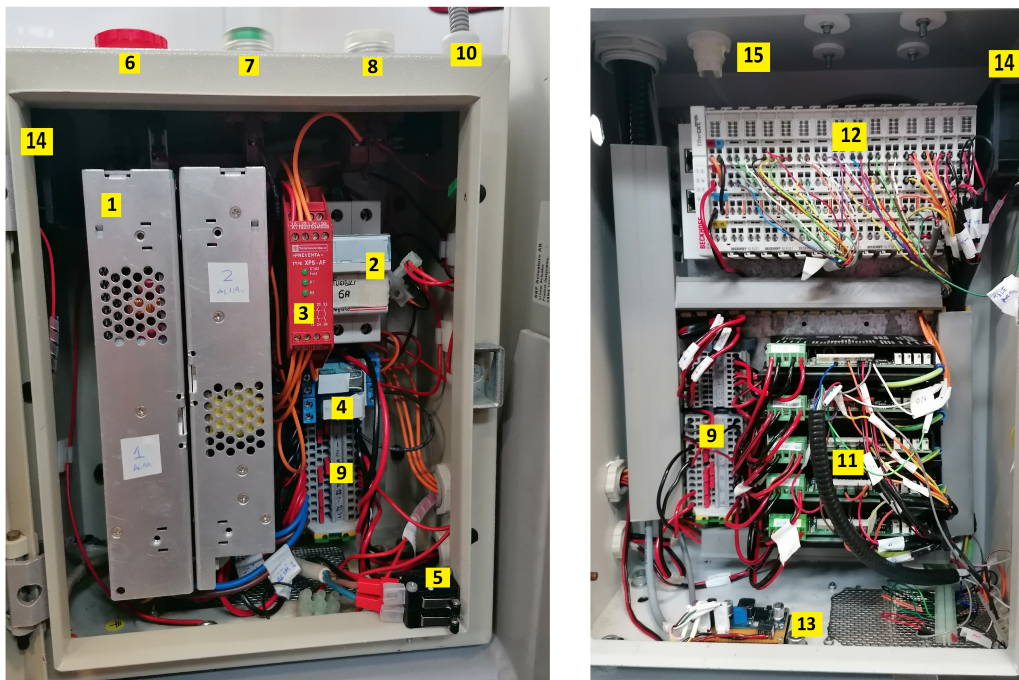


Figura 3.1: Interno dei quadri elettrici alla base del Marisa

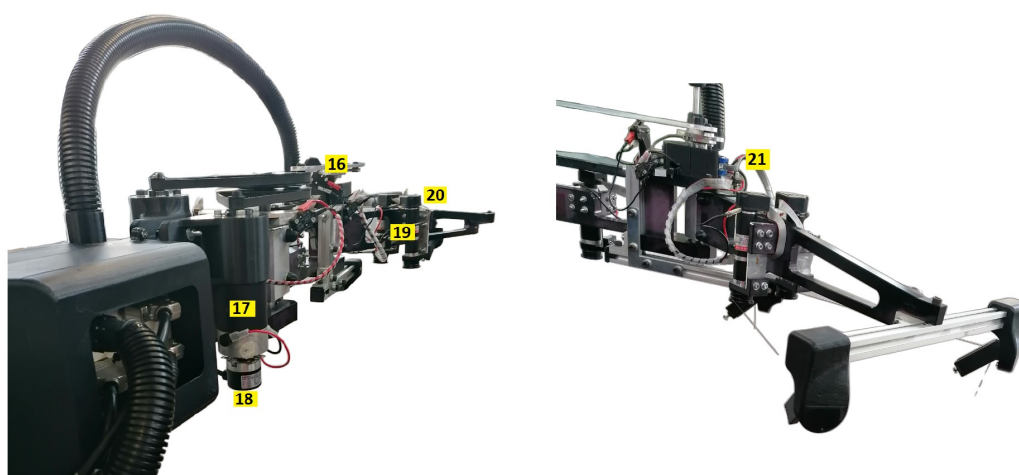


Figura 3.2: Componenti elettrici lungo il braccio seriale

Nel caso in cui ciò non fosse possibile, allora sarà necessario andare ad interrompere il circuito, evitando danni potenzialmente anche gravi, sia per la macchina che per il paziente.

Di seguito si riportano i componenti elettronici utilizzati a tale scopo.

1. Alimentatori Skynet SNP-B209

Erogando 24 V in continua partendo dai 220 V in alternata provenienti dalla rete, forniscono alimentazione all'intero sistema. Nel dettaglio, è possibile erogare, attraverso il collegamento in serie dei due alimentatori descritto più avanti in questo capitolo, due tipi di alimentazione: una a 24 V e una a 48 V. La necessità di avere a disposizione due diversi livelli di tensione deriva dal fatto che i vari componenti elettrici non hanno lo stesso assorbimento, in particolare:

- **Alimentazione a 24 V**

ventole di raffreddamento (una per ogni quadro), driver per motori dei cavi, moduli Beckhoff, scheda di controllo sensori induttivi (poi ridotta a 5 V), lampadina pulsante di riarmo e lampada di macchina armata.

- **Alimentazione a 48 V**

driver per motori dei link.

Per i calcoli che hanno portato alla scelta di tali alimentatori, si rimanda alla sezione 3.3.

2. Fusibili di sicurezza a 6 A

Dispositivi di sicurezza che proteggono il circuito elettrico totale, sacrificandosi nel caso in cui si verifichino sovraccarichi di corrente provenienti dalla rete superiori al valore di 6 A. Questi fenomeni potrebbero risultare molto dannosi e pericolosi, pertanto l'ausilio di fusibili rende più sicuro il sistema, in quanto si ha una rottura localizzata e concentrata, facilmente identificabile e la cui riparazione richiede soltanto la sostituzione di questi

elementi, i cui costi sono decisamente ridotti.

3. Relè Telemecanique XPS-AF5130

Relè utilizzato per la gestione dell'arresto di emergenza e della fase di armo/disarmo dei driver che erogano potenza ai motori nei casi in cui venga premuto uno dei contatti di sicurezza distribuiti in vari punti del manipolatore oppure qualora venisse riscontrato qualche malfunzionamento di natura elettrica. Si riportano di seguito le caratteristiche di tale dispositivo, il cui funzionamento all'interno del circuito di funzionamento complessivo sarà spiegato oltre.

Modello MODULO SICUREZZA	textbfXPSAF5130
Applicazione modulo sicurezza	Per arresto d'emergenza e controllo commutazione
Funzione del modulo	Monitoraggio arresto di emergenza 1 canale Monitoraggio arresto di emergenza 2 vie Monitoraggio protezione mobile
Livello di sicurezza	Può raggiungere PL e/categoria 4 conforme a EN/ISO 13849-1 Fino alla categoria 3 conforme a EN/IEC 62061
Dati di affidabilità sicurezza	DC > 99% conforme a EN/ISO 13849-1 MTTFd = 243 anni conforme a EN/ISO 13849-1 PFHd = 4.62E-9 1/h conforme a EN/IEC 62061
Tipo di avviamento	Configurabile
Resistenza linea	90Ω
Potere di interruzione	180 VA in attesa AC-15 C300 uscita relè 1800 VA picco AC-15 C300 uscita relè
Potere di interruzione	1,5 A a 24 V (DC-13) costante del tempo: 50 ms per uscita relè
Corrente termica uscita	6 A per relè per uscita relè
Corrente termica convenzionale in aria libera [Ith]	18 A
Calibro del fusibile associato	4 A gG o gL per uscita relè conforme a EN/IEC 60947-5-1, DIN VDE 0660 parte 200 6 A ad intervento per uscita relè conforme a EN/IEC 60947-5-1, DIN VDE 0660 parte 200
Corrente di uscita minima	10 mA per uscita relè
Tensione di uscita minima	17 V per uscita relè
Massimo tempo di risposta all'apertura dell'ingresso	40 ms
Tensione nominale di isolamento [Ui]	300 V (grado di inquinamento 2) conforme a IEC 60947-5-1 300 V (grado di inquinamento 2) conforme a DIN VDE 0110 parte 1
Tensione nominale di tenuta ad impulso [Uimp]	4 kV categoria sovratensione III conforme a IEC 60947-5-1 4 kV categoria sovratensione III conforme a DIN VDE 0110 parte 1
Segnalazione locale	3 LED
Assorbimento di corrente	30 mA a 24 V CA sull'alimentazione elettrica

Figura 3.3: Tabella dati e funzionamento Relè Telemecanique XPS-AF5130

4. Relè Finder type 95.75

Due relè identici il cui compito è operare sempre nel contesto del circuito di sicurezza attraverso l'apertura e la chiusura dei loro contatti interni, di tipo NO ¹, NC ², COM ³. Come si vedrà oltre, questi costituiscono i due canali per il relè XPS-AF5130 nel caso di *monitoraggio arresto di emergenza a 2 vie*.

5. Interruttore NO di sicurezza

Ottenuto con un microinterruttore a rullo, fa sì che l'alimentazione possa essere resa disponibile solamente quando la porta del quadro elettrico è chiusa (contatto chiuso).

6. Pulsante di emergenza 1: da quadro

Uno dei tre pulsanti di emergenza presenti in MariBot 2.0 che permettono di tagliare manualmente l'alimentazione ai driver in caso di necessità. Questi sono implementati con dei contatti NC e, grazie all'azione del circuito di sicurezza ottenuto sfruttando i relè "2" e "3", è sufficiente una pressione istantanea per bloccare il flusso di corrente. Per ripristinarlo, è necessario premere il pulsante di riarmo "8". In caso di riarmo avvenuto correttamente, si accende la lampada verde "7".

7. Luce di conferma: sistema armato

Luce accesa se il sistema è armato correttamente e pronto per l'esecuzione del programma o attualmente in esecuzione.

8. Pulsante di riarmo

Luce gialla eccesa indica che il sistema è armato oppure può essere riarmato perchè i contatti di emergenza sono stati ripristinati (se così non fosse, la

¹Normally Open

²Normally Cloded

³Common

luce sarebbe spenta).

9. **Morsettiere per il ricircolo del segnale elettrico**

Di dimensione variabile, permettono di gestire le tensioni prodotte dagli alimentatori facendole fluire da un quadro all'altro e verso il braccio seriale, connettendo in maniera adeguata i vari componenti.

10. **Collegamento al pulsante di emergenza 2: da colonna** Cablaggio che porta al pulsante di emergenza posizionato sulla colonna del Marisa. Funziona in modo analogo a "6".

16. **Contatti di finecorsa** Pulsanti posizionati nei punti che determinano il range massimo di movimento dei due link, definendo quindi lo spazio di lavoro planare del robot. Essi vengono inseriti per garantire che il circuito venga interrotto ogni qualvolta si cerchi di ottenere un movimento al di fuori del range possibile, con conseguenti urti tra le diverse parti meccaniche. Questi contatti sono ottenuti con quattro pulsanti NO.

Gli elementi appena descritti concorrono alla realizzazione del circuito complessivo di alimentazione e di sicurezza in bassa tensione. Nel rispetto della Norma IEC 60601-1 riguardante la sicurezza dei dispositivi elettromedicali, si rende necessario implementare un sistema che permetta di tagliare l'alimentazione fornita dagli alimentatori, cioè a 24V e 48V.

Le operazioni principali che devono essere svolte sono due:

- **Armatura dei driver**

Una volta acceso il quadro attraverso l'interruttore di ON/OFF sul lato del quadro alla sinistra della base del Marisa, si accende la luce bianca posta sulla sommità dello stesso: questa condizione definisce la situazione in cui i driver sono alimentati (luce rossa accesa su ognuno di essi) ma non abilitati a fornire alimentazione al motore da essi comandato. Per passare in questa

seconda fase, sarà necessario premere il medesimo pulsante bianco (fase di armatura del sistema), cui consegnerà l'accensione della lampadina verde del quadro, oltre che di quelle proprie dei cinque driver.

Il passaggio "driver non alimentati - driver alimentati" è reso possibile attraverso il collegamento tra il circuito di sicurezza dei relè nel quadrom di sinistra e i contatti di INHIBIT dei diversi driver nel quadro di destra.

In questa condizione i motori sono effettivamente alimentati dalla corrente erogata dai driver e pertanto è possibile avviare l'esecuzione del programma di riabilitazione.

Occorre evidenziare che è stato reso necessario ottenere anche una tensione di alimentazione a 48 V per i motori dei link, realizzata tramite la serie dei due alimentatori.

- **Interruzione circuito in caso di emergenza**

Il circuito di sicurezza ha il fondamentale compito di interrompere il flusso di corrente tutte le volte che si verifichi un'anomalia oppure quando il terapeuta ritenga sia necessario arrestare l'esecuzione per un malfunzionamento durante l'esercizio. Questo, quindi, si verifica con la pressione di uno dei diversi pulsanti di emergenza, dell'apertura della porta del quadro elettrico (e relativa commutazione del contatto NC) al raggiungimento dei finecorsa dei link oppure in qualsiasi altro caso di malfunzionamento di natura elettrica (sovraccarichi, ecc.). In tutti questi casi, dunque, si interromperà il circuito di comando, che comunque rimarrà acceso grazie ai relè impiegati, ma porterà alla condizione controllata di sistema non armato (sola luce binaca accesa, driver alimentati ma non abilitati, con luce rossa). Per poter nuovamente tornare nella condizione di esecuzione del programma occorre, ovviamente, riarmare il sistema come descritto al punto precedente.

Si riportano di seguito due schemi parziali che nel complessivo descrivono i collegamenti effettuati che consentono l'adempimento delle funzioni sopra descritte; i circuiti realizzati nel quadro alla sinistra di Marisa, contenente quindi gli alimen-

tatori e i relè di sicurezza.

La logica che sta alla base di tali schemi è fortemente dipendente dal relè XPS-AF5130, che di fatto rappresenta il centro di controllo per quanto riguarda l'intero circuito di emergenza.

Questo genere di dispositivo offre diverse modalità per il monitoraggio e arresto di un impianto a 24 V, conta fino a 2 canali di uscita e per ciascun canale sono presenti due interruttori (rappresentati dai terminali NO dei relè Finder) a scopo ridondante, monitorati continuamente nel loro stato. Una volta alimentato e configurato nella modalità desiderata per la gestione del quadro, nel caso in esame di *"monitoraggio arresto di emergenza 2 vie"* il relè ha il compito di controllare continuamente lo stato di due suoi canali a 24 V. Le situazioni che possono verificarsi sono le seguenti:

1. tra ingresso ed uscita è riscontrata una differenza di potenziale intorno agli 0 V (condizione di cortocircuito). In questo caso il relè ritiene che non ci siano anomalie in corso ed ha quindi il compito di tenere i suoi contatti in uscita chiusi.
2. in uno dei canali di monitoraggio viene rilevato:
 - un aumento della resistenza lungo il circuito
 - un'interruzione del circuito a causa di una tensione rilevata ai capi dei canali di monitoraggio che sia:
 - superiore ad una soglia di guardia
 - tensione a vuoto (circa 24 V)

A questo punto il relè riapre i contatti dedicati al canale in uscita 1 e 2, di fatto interrompendo la continuità elettrica dei due. Una volta risolta la condizione che ha comportato lo stato di anomalia nei canali di monitoraggio, il relè di sicurezza richiede una condizione aggiuntiva di sicurezza perchè riavvenga la chiusura dei contatti N.O. dei canali 1 e 2. Questa condizione è la cortocircuitazione di un terzo canale (*RESET*) di monitoraggio destinato unicamente al riarmo del sistema.

Il relè XPSAF ha inoltre 3 spie luminose che utilizza per segnalare all'operatore lo stato dei contatti in uscita del relè e dei 2 canali monitorati.

L'abilitazione dei driver (e quindi la possibilità di avviare il motore) può solo in seguito alla pressione del pulsante di riarmo bianco. Questo infatti eccita la bobina del relè XPS, i cui contatti chiudono la linea dedicata al terzo canale di monitoraggio *RESET*. Se non è stato premuto il fungo di emergenza sulla pulsantiera, allora si hanno la chiusura dei due interruttori N.O. interbloccanti presenti nel Canale 1 e 2, rappresentati dai contatti dei relè Finder: si alimenteranno le loro bobine in modo da poter ottenere l'inibizione dei driver. Si può notare in questa configurazione che nel caso in cui dovesse avvenire un guasto in uno dei contatti N.C., durante la loro fase di contatti aperti si crea una condizione di anomalia tale da poter inibire ancora il motore. Ciò è possibile in quanto l'altro contatto ancora funzionante può ancora, da solo, fungere da cortocircuito e perseguire la logica richiesta.

Si evidenzia che l'aggiunta dei contatti normalmente chiusi, appartenenti ai relè Finder nei due canali dell'XPS evitano che il canale "RESET" venga inutilmente stimolato quando non necessario (fase di marcia del motore).

3.1.2 Quadro di potenza e controllo

La parte di controllo rappresenta il cervello di MariBot 2.0. Essa si costituisce di due componenti principali. La prima è rappresentata dai moduli Beckhoff, che garantiscono la gestione del manipolatore in *Real-Time* grazie al protocollo di comunicazione EtherCAT. Secondo elemento della parte di controllo è dato dall'insieme dei cinque driver di alimentazione dei motori.

11. Driver AMC25A8

Driver appositamente progettati per il controllo di motori in corrente continua di tipo brushless a elevata frequenza di commutazione. Nella versione originale del MariBot erano usati solo per il controllo dei motori dei link, ora invece ce n'è uno per ogni motore (cavi compresi). Per poter essere sfruttati al pieno delle proprie potenzialità, è necessario che questi dispo-

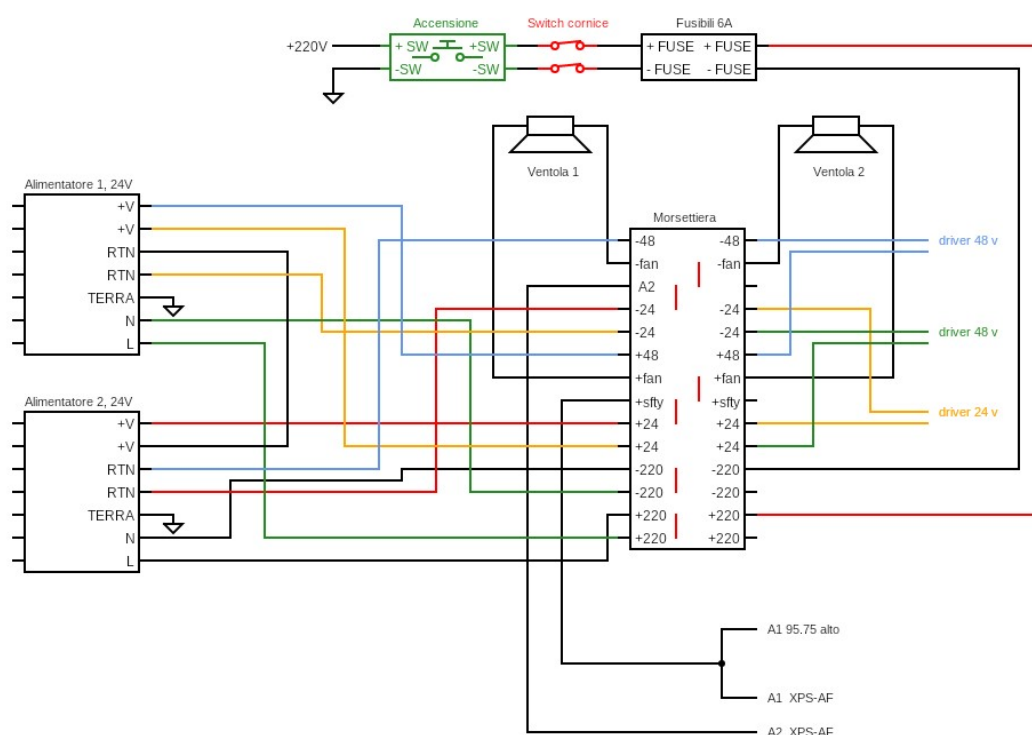


Figura 3.4: Schema parziale del circuito di sicurezza e di alimentazione, parte 1

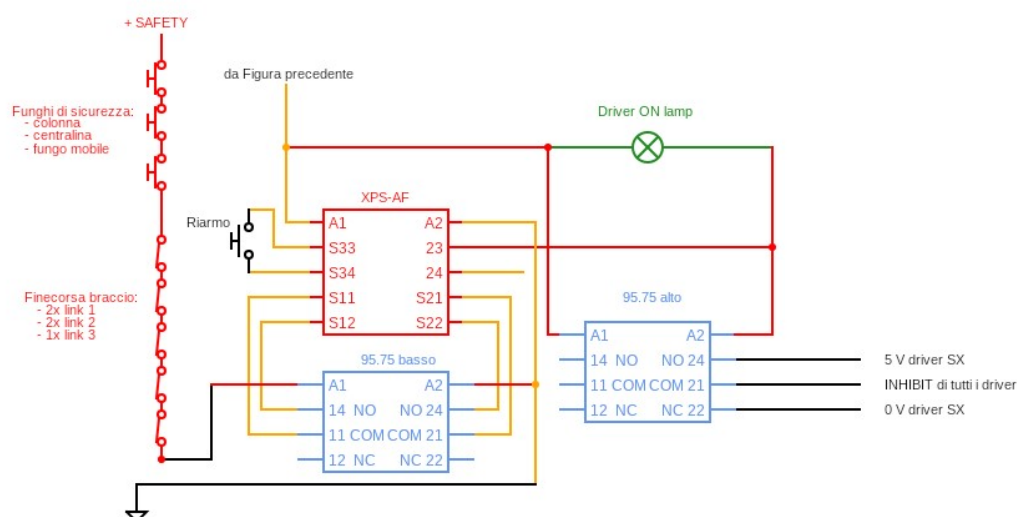


Figura 3.5: Schema parziale del circuito di sicurezza e di alimentazione, parte 2

sitivi vengano adeguatamente configurati. Nel caso in esame, per esempio, occorre impostare come modalità di funzionamento il "current mode": questa modalità si ripercuote sul governare la coppia erogata dai motori ad ogni istante, attraverso la gestione della corrente di armatura prodotta dalla logica di controllo. Per imporre il funzionamento in current mode occorre che solo il terzo interruttore degli switch presenti sul driver sia ON.

Per terminare la configurazione dei driver bisogna poi regolarne i quattro potenziometri: il primo (LOOP GAIN), dev'essere ruotato in verso antiorario fino al suo limite, il terzo (REF IN GAIN) nel verso opposto, sempre fino a finecorsa. La regolazione del secondo potenziometro (CURR LIMIT) risulta un po' più laboriosa, in quanto richiede l'imposizione della relazione tra corrente e tensione massima.

Partendo quindi dai dati del motore cui il driver fa riferimento (forniti da data sheet), si possono ricavare i valori di coppia continuativa massima C_{max} e della costante di coppia K_t . Secondo la relazione

$$i_{max} = \frac{C_{max}}{K_t} \quad (3.1)$$

sarà quindi possibile ottenere la corrente di armatura massima i_{max} , erogabile quando viene fornita una tensione di controllo massima, a valle di limiti di saturazione, che chiameremo V_{sat} . Erogando quindi V_{sat} (o, per garantire maggior sicurezza durante la configurazione, un rapporto della stessa, per esempio un centesimo), si dovrà misurare una corrente in uscita dal driver corrispondente proprio a i_{max} (o allo stesso rapporto usato per la tensione). Il quarto e ultimo potenziometro (TEST/OFFSET) è quello che determina l'offset di corrente che il driver eroga quando ad esso è fornita una tensione di controllo nulla. Nel caso in esame occorre annullare questo valore, pertanto fornendo una tensione nulla si dovrà ruotare il potenziometro 4 fino a quando anche la corrente in uscita misurata non sarà pari a zero.

12. Moduli di controllo Beckhoff

Tutti i segnali provenienti dai diversi componenti elettronici finora descrit-

ti confluiscono in una serie di moduli di controllo Beckhoff ⁴ e, insieme al programma realizzato in ambiente *MATLAB* e *SimulinkRealTime*, governano il funzionamento del robot.

I moduli impiegati all'interno di MariBot 2.0 si possono distinguere in quattro tipi e svolgono funzioni diverse:

- **EK1100: EtherCat Coupler**

Modulo di accoppiamento tra il protocollo EtherCAT e i terminali (PC Guest e PC Host).

- **EL5101: 1-channel encoder interface**

Modulo per la gestione mirata di segnali provenienti da encoder. Nel caso del MariBot 2.0 si usano quindi cinque di questi moduli, uno per encoder, ordinati come segue, da sinistra verso destra: cavo SX, cavo DX, cavo BACK, LINK1, LINK2.

- **EL4134: 4-channel analog output**

Modulo che genera la tensione d'uscita nel range $[-10, +10]$ V. Può essere considerato come un modulo di "scrittura". La risoluzione è garantita da un quantizzatore a 16 bit.

- **EL3004: 4-channel analog input**

Modulo che legge la tensione proveniente dai motori. Può essere considerato come un modulo di "lettura". La risoluzione è garantita da un quantizzatore a 12 bit.

Per adempiere alle loro mansioni e fornire, ad ogni istante, la corretta alimentazione agli attuatori, è necessario che i driver scambiano segnali di controllo con i moduli Beckhoff.

⁴Per ulteriori informazioni: www.beckhoff.com

Tra i segnali interessati ci sono quelli che arrivano dai relè Finder del quadro di sinistra i quali vanno a gestire la parte di INHIBIT dei driver in base alle risposte del circuito di sicurezza.

Per quanto concerne il controllo eseguito dei moduli Beckhoff, invece, questo si può distinguere in tre parti, ciascuna eseguita da una tipologia precisa di modulo: la parte di lettura encoder (gestiti dai moduli EL5101), la parte di lettura della corrente (moduli EL3001) e la parte di scrittura della tensione, dunque della corrente, per azionare i motori (moduli EL4134).

13. **Sensori induttivi**

Sensori induttivi (uno per ogni membro del braccio) impiegati in fase di allineamento dei link: permettono di memorizzare la posizione corrispondente allo zero encoder, che si verifica quando il sensore rileva la variazione di induttanza data dal posizionamento, al di sotto dello stesso, dell'acciaio che costituisce la struttura meccanica del braccio. Il segnale così generato viene gestito da "13".

3.2 **Sensori e altri elementi**

Ulteriore serie di componenti impiegati nel robot che svolgono funzioni al di fuori di quelle precedentemente descritte.

14. **Ventole di raffreddamento**

Alimentate da tensione a 24 V in continua, raffreddano ciascuno dei due quadri elettrici.

15. **Collegamento al telecomando terapeuta**

Il telecomando terapeuta è realizzato per consentire al terapeuta di avere il controllo diretto dei principali comandi utili durante l'esercizio, cioè la possibilità di acquisire i punti di passaggio e commutare tra le varie fasi

della riabilitazione (per la descrizione precisa si rimanda al capitolo 3) oltre che la possibilità di terminare l'esecuzione in caso di emergenza. A tal fine è stato realizzato tramite stampa 3D un como telecomando con un pulsante verde monostabile NO (più avanti definito come "pulsante di acquisizione") e un classico fungo di emergenza NC.



Figura 3.6: Telecomando terapeuta

3.3 Azionamenti

Il movimento del manipolatore è gestito, per ogni cavo e per ogni link, da un motore elettrico di tipo brushless, controllato attraverso i moduli Beckhoff per erogare la giusta potenza in ogni istante; a ciascun motore corrisponde poi un encoder, dispositivo in grado di misurare la posizione attuale del rotore e che permette, conseguentemente, di determinare la posizione del manipolatore nello spazio di lavoro.

21. Scheda di gestione dei sensori induttivi

Scheda di gestione dei segnali provenienti dai segnali induttivi "21", impiegati in fase di allineamento del braccio per memorizzare la posizione corrispondente allo zero encoder.

Tali sensori, capaci di fornire un segnale qualora rilevino il passaggio di un materiale ferromagnetico in loro prossimità, sono localizzati uno per link e vengono

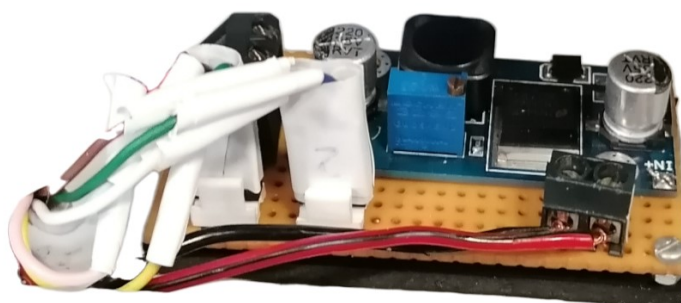


Figura 3.7: Scheda di controllo dei sensori induttivi

gestiti dalle periferiche Beckhoff EL3004.

Dal link, infatti, insieme a tutti i diversi cavi utilizzati nei cablaggi di alimentazione e controllo encoder descritti in precedenza, scendono tre terminali per ogni induttivo, due dei quali di alimentazione e uno invece di segnale. Questi entrano in principio nella scheda di controllo riportata in precedenza, il cui compito fondamentale è quello di adattare la tensione erogata dagli alimentatori a 24 V ai 5 V richiesti dai sensori e fornire in uscita il segnale utilizzabile dalla centralina Beckhoff. Per chiarezza si riporta uno schema che rappresenta i terminali dei moduli Beckhoff impiegati per la gestione dei segnali provenienti dai sensori induttivi e dal pulsante di acquisizione del telecomando terapisti.

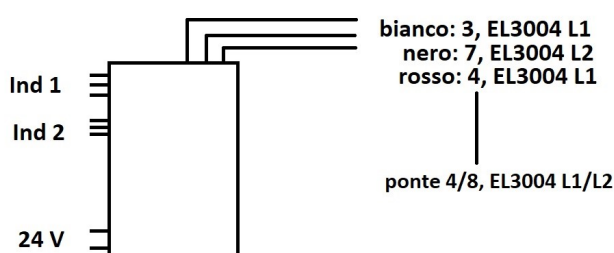


Figura 3.8: Scheda di controllo e segnali Beckhoff

17. Motori link: MAE M543-0900

Motori in continua a magneti permanenti usati per muovere LINK1 e LINK3, lato del quadrilatero articolato che rimanda al movimento di LINK2. se ne riportano di seguito le caratteristiche.

MAE M543-0900	
C_{picco}	1.44 Nm
C_{nom}	0.4 Nm
K_E	9 V/Krpm
K_T	0.086 Nm/A
R	1.1 Ω
velocità nominale n	800 rpm

18. Encoder link: Baumer electric BHK 16.05A2000

Encoder incrementale per la misura della posizione dei motori dei link.



Figura 3.9: Motore MAE M543-0900 e encoder Baumer electric BHK 16.05A2000

19. Motori cavi: Maxon RE25 e Maxon RE26

Motori impiegati per il movimento dei tre cavi per il movimento dell'ortesi. Nel dettaglio si utilizzano due RE25 per i cavi anteriori (più avanti chiamati semplicemente "cavo SX" e "cavo DX") e un RE26 per il posteriore ("cavo BACK").

20. Encoder cavi: HEDL5540

Encoder incrementale per la misura della posizione dei motori dei cavi.

Maxon RE25		Maxon RE26	
K_E	1.447 V/rpm	K_E	1.407 V/rpm
K_T	21.3 mNm/A	K_T	23.4 mNm/A
R	2.27 Ω	R	2.32 Ω
η	0.75	η	0.7
raggio puleggia r	15 mm	raggio puleggia r	15 mm
rapp. trasmissione i	33	rapp. trasmissione i	29



Figura 3.10: Motori Maxon RE25, Maxon RE26 e encoder HEDL5540

3.4 Dimensionamento del quadro elettrico

Una volta descritti i componenti adottati e noti i loro parametri caratteristici, si riportano di seguito i calcoli svolti che hanno portato alla loro scelta.

Si analizzano quindi i carichi richiesti col fine di individuare i campi di funzionamento dei motori.

Questo è stato possibile effettuando una serie di simulazioni che hanno permesso di determinare le coppie richieste in fase di esercizio, in modo tale da poter determinare conseguentemente le coppie motrici necessarie. Su queste si è poi basata la scelta dei motori da impiegare per l'azionamento di link e cavi.

3.4.1 Dimensionamento elettrico dei motori dei link

Si inizia con lo studio dell'assorbimento dei motori dei link, suddividendolo nei casi di erogazione della coppia nominale (fornita alla velocità di rotazione nominale n) e di picco (a rotore fermo).

$$I_{\text{nom}} = \frac{C_{\text{nom}}}{K_T} = 4.65, \text{ A} \quad (3.2)$$

$$V_{nom} = R \cdot I_{nom} + K_E \cdot n = 12.3V \quad (3.3)$$

$$I_{picco} = \frac{C_{picco}}{K_T} = 16.74A \quad (3.4)$$

$$V_{picco} = R \cdot I_{picco} + K_E \cdot n = 18.7V \quad (3.5)$$

Noti poi i parametri di alimentazione massima dei driver forniti da datasheet (osservando in particolare la tensione continua di alimentazione, compresa tra 20-80V e la massima corrente d'uscita $I_{out}=12.5A$), si evince che le richieste della condizione di picco andrebbero ben oltre tali limiti. Ciò richiederebbe l'impiego di un gruppo di alimentazione con capacità superiori, che però risulterebbe sovradimensionato nel caso di impiego normale. A tal fine si è deciso di utilizzare come valori di riferimento per la scelta dei componenti solo quelli nominali.

3.4.2 Dimensionamento elettrico dei motori dei cavi

Si prosegue verificando il dimensionamento dei motori dei cavi, distinguendoli tra "anteriori" (RE25) e "posteriore" (RE26). I driver adottati, com'è noto, sono i medesimi impiegati per i motori dei giunti.

- **Motori per cavi anteriori**

In base ai dati tecnici riportati e alle leggi note di un motore in continua, tenuto poi conto della velocità massima ammessa dal riduttore (8000 rpm da datasheet), sono state eseguite delle prove sperimentali durante le quali il motore è stato alimentato con il massimo voltaggio consentito dal motore (24V). In queste condizioni è stato possibile constatare in primis che il riduttore raggiunge la propria velocità massima e tenderebbe a superarla (9300 rpm): ciò rende necessaria l'implementazione di un controllore di velocità a limitarla.

In queste condizioni, la velocità lineare di avvolgimento (o srotolamento) del cavo è pari a 433 mm/s. Si osserva inoltre che la corrente necessaria per esercitare la massima coppia continua (21.6 mNm) è pari a:

$$I_{\text{nom}} = \frac{C_{\text{nom}}}{K_T} = 988.7\text{mA} \quad (3.6)$$

mentre la tensione nominale sarà pari a:

$$V_{\text{nom}} = R \cdot I_{\text{nom}} + K_E \cdot n = 11.6V \quad (3.7)$$

- **Motore per cavo posteriore**

Si ripete quanto appena descritto anche per il motore R26 posteriore, il quale ha una velocità massima dal riduttore di 6000 rpm. Le prove svolte hanno in questo caso portato i seguenti risultati: il riduttore raggiunge e supera la propria velocità massima (9712 rpm) se non si prevede un anello di velocità nel controllore; il cavo conseguentemente si muove a 286 mm/s; la corrente necessaria per esercitare la massima coppia continua (30.7 mNm) è pari a 1.31 A, mentre la tensione nominale è pari a 8.5V.

3.4.3 Scelta degli alimentatori

Una volta determinato l'assorbimento dei motori, è necessario scegliere le sorgenti di alimentazione capaci di erogare la potenza necessaria, le quali dovranno inoltre garantire il funzionamento degli altri dispositivi elettronici presenti nel manipolatore: lampadine dei pulsanti di riarmo e di marcia, ventole dei quadri elettrici, sensori induttivi e relativa scheda e moduli Beckoff.

Si verifica, quindi la scelta dei due alimentatori Skynet SNP-B209, impiegati in modo che il collegamento in serie dei due alimentatori sia in grado di fornire sia l'alimentazione richiesta dei motori dei link (48V) sia l'alimentazione richiesta dai motori dei cavi (24V). Si verifica inoltre la potenza totale erogata quando si collegano in serie i due alimentatori, così da poter alimentare correttamente tutti i vari componenti.

- **Alimentazione motori dei link**

Si considera la tensione nominale di ciascun motore calcolata in precedenza (6.02V) e la corrente richiesta per il funzionamento nominale dei motori

(pari a 4.65A per ogni driver). La potenza elettrica assorbita da entrambi i motori sarà pari a:

$$P_{el,link} = 2 \cdot V_{nom,link} \cdot I_{nom,link} = 56W \quad (3.8)$$

- **Alimentazione motori dei cavi**

In modo analogo a quanto fatto precedentemente, si considera la tensione nominale e la corrente nominale dei motori anteriori (11.6 V e 1.0 A) e di quelli posteriori (8.5 V e 1.31 A) e se ne calcolano le rispettive potenze elettriche assorbite (23.2 W per entrambi i motori anteriori e 11.1 W per il motore posteriore).

Si può determinare l'assorbimento totale come somma dell'assorbimento dei singoli componenti, risultando in un valore di potenza elettrica pari a:

$$P_{el,tot} = P_{el,link} + P_{el,cavi} = 91W \quad (3.9)$$

Dal momento che ciascun alimentatore può fornire una potenza elettrica pari a 200 W, il dimensionamento elettrico si considera ampiamente verificato entro i limiti di potenza a disposizione e, pertanto, è possibile omettere la stima della potenza assorbita degli altri componenti elettrici presenti.

3.5 Cablaggio

Ci si appresta ora a descrivere il cablaggio realizzato per il funzionamento di MariBot2.0.

Come si è potuto osservare dalle sezioni precedenti, il robot richiede l'impiego svariate componenti elettroniche le quali devono necessariamente essere interconnesse correttamente per poter eseguire il loro compito.

Avendo già descritto la parte di alimentazione e sicurezza nella sezione precedente, che tra le altre cose permette di fornire il passaggio della tensione dal quadro sinistro al quadro destro alla base del Marisa, dunque nella sezione dei driver, occorre spiegare come queste riescano ad arrivare fino ai motori sfruttando determinati connettori e al contempo di quali siano i segnali di comando adottati.

Dallo stesso quadro dipartono anche i cavi di segnale inerenti agli encoder e ai sensori induttivi. Vista la complessità dei collegamenti realizzati, si preferisce anche in questo caso approcciare alla descrizione per blocchi in base alla funzione che quella determinata parte di circuito deve soddisfare.

Innanzitutto si riporta in modo schematico il complessivo dei collegamenti realizzati su MariBot2.0.

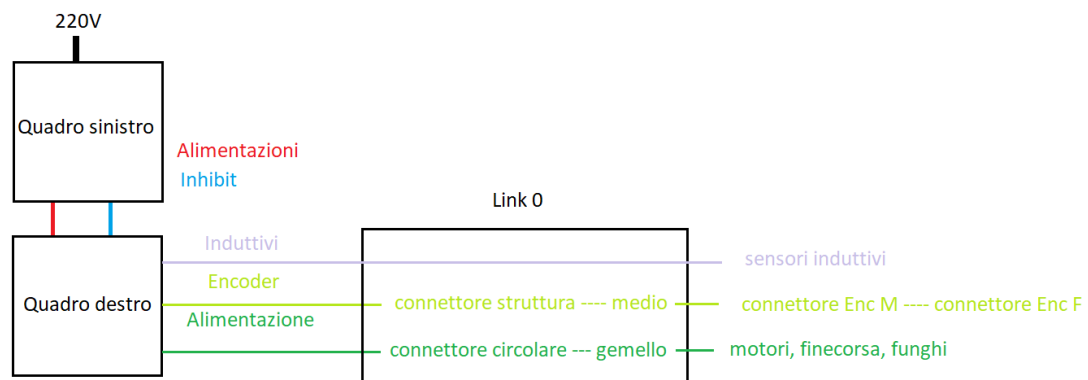


Figura 3.11: Schema generale dei collegamenti nel robot

3.5.1 Collegamenti di alimentazione e finecorsa

Il cablaggio relativo al circuito di alimentazione, ampiamente descritto nelle sezioni precedenti, deve necessariamente terminare ai poli dei motori dei giunti e dei cavi. Per fare ciò, attraverso una canalina, i vari cavi positivi e negativi salgono fino al LINK0 del manipolatore dove confluiranno nel connettore circolare di seguito riportato. Con un connettore gemello dall'altro lato della struttura, poi, dipartiranno per connettersi in maniera adeguata ai diversi motori, ai finecorsa situati nei punti estremi del range di movimento dei link e ai diversi funghi di emergenza, secondo i collegamenti riportati in precedenza.

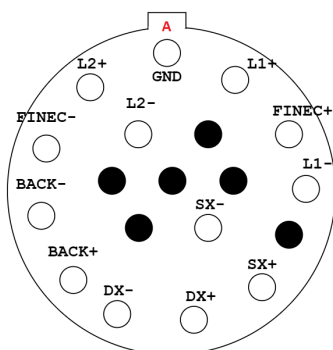


Figura 3.12: Connettore che permette il collegamento dal quadro di alimentazione di destra ai motori, finecorsa sul braccio seriale e ai vari funghi di emergenza

3.5.2 Collegamenti driver

Per poter comprendere meglio il funzionamento delle connessioni inerenti i driver, occorre partire con la descrizione dei diversi pin e connettori degli stessi, descritti nel relativo datasheet.

Si premetta che il collegamento della parte di alimentazione proveniente dal quadro di sinistra è effettuato in maniera abbastanza elementare sfruttando le morsettiere verdi ricavati sugli stessi.

Tra tutti i pin a disposizione, numerati da 1 a 16, si impiegheranno solo i pin 1, 2, 4, 5, 8 e 11, come specificato nelle tabelle che seguiranno.

La logica alla base del circuito è che (salvo eccezioni comunque descritte) i terminali 4 e 5 siano adibiti alla scrittura, pertanto connessi ai moduli EL4134; i connettori 2 e 5 invece si occupano della lettura (collegati ai moduli EL3004) mentre i connettori 11 sono connessi a un circuito, realizzato tramite un mammut elettrico, che mette in parallelo tra loro tutti i segnali di INHIBIT, alimentati dal relè Finder 95.75 più in alto nel quadro. Il pin numero 1 è impiegato solo dai driver SX e DX che lo usano per la gestione dei segnali provenienti dal telecomando e dal relè nominato pocanzi.

Occorre precisare che i pin dei diversi moduli Beckhoff sono impiegati in modo che il modulo più a destra sia adibito ai soli link, mentre quello di sinistra ai soli cavi.

CONNECTOR	PIN	NAME	DESCRIPTION / NOTES	I/O
P1	1	+5V OUT	Internal DC-to-DC converter, outputs regulated voltages of $\pm 5\text{ V}$ @ 3 mA for customer use. Short circuit protected.	O
	2	SIGNAL GND		GND
	3	-5V OUT		O
	4	+REF IN	Differential analog input, maximum $\pm 15\text{ V}$, 50K input resistance.	I
	5	-REF IN		
	6	-TACH IN	Maximum $\pm 60\text{ VDC}$, 60K input resistance.	I
	7	+TACH (GND)		
	8	CURRENT MONITOR OUT	This signal is proportional to the actual current in the motor leads. Scaling is 2A/V for 12A8 and 4 A/V for 25A8, 20A14 and 20A20.	O
	9	CURRENT REFERENCE OUT	Command signal to the internal current-loop. The maximum peak current rating of the amplifier always equals 7.25V at this pin. See current limit adjustment information below.	O
	10	CONTINUOUS CURRENT LIMIT	Can be used to reduce the factory-preset maximum continuous current limit.	I
	11	INHIBIT	This TTL level input signal turns off all four power devices of the "H" bridge drive when pulled to ground. This inhibit will cause a FAULT condition and a red LED. For inverted inhibit inputs; see section "G".	I
	12	+INHIBIT	Disables the amplifier for the "+" direction only. This inhibit will not cause a FAULT condition or a red LED.	I
	13	-INHIBIT	Disables the amplifier for the "-" direction only. This inhibit will not cause a FAULT condition or a red LED.	I
	14	FAULT OUT (red LED)	TTL compatible output. It becomes high during output short-circuit, over-voltage, over-heating, inhibit, and during "power-on reset". Fault condition indicated by a red LED.	O
	15	NC	Not connected	
	16			

Figura 3.13: Connettori driver da datasheet

Pin Driver	EL4134	EL3004	Altre connessioni
1	-	-	NO 24, 95.75 alto
2	-	-	Serie INHIBIT
4	1	-	-
5	2	-	-
8	-	1	-
11	-	-	Serie INHIBIT

Figura 3.14: Tabella collegamenti driver CAVO SX

Pin Driver	EL4134	EL3004	Altre connessioni
1	-	-	Telecomando (grigio)
2	-	6	-
4	5	-	-
5	6	-	-
8	-	5	-
11	-	-	Serie INHIBIT

Figura 3.15: Tabella collegamenti driver CAVO DX

Pin Driver	EL4134	EL3004	Altre connessioni
1	-	-	-
2	-	4	-
4	3	-	-
5	4	-	-
8	-	3	-
11	-	-	Serie INHIBIT

Figura 3.16: Tabella collegamenti driver CAVO BACK

Pin Driver	EL4134	EL3004	Altre connessioni
1	-	-	-
2	-	2	-
4	1	-	-
5	2	-	-
8	-	1	-
11	-	-	Serie INHIBIT

Figura 3.17: Tabella collegamenti driver LINK1

Pin Driver	EL4134	EL3004	Altre connessioni
1	-	-	-
2	-	6	-
4	1	-	-
5	2	-	-
8	-	5	-
11	-	-	Serie INHIBIT

Figura 3.18: Tabella collegamenti driver LINK2

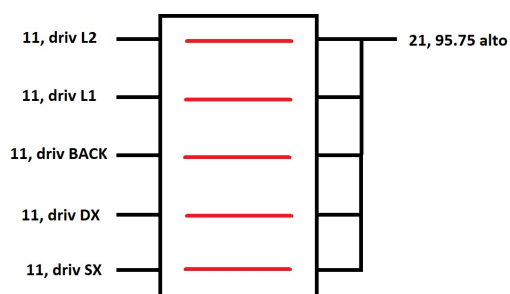


Figura 3.19: Mammut di collegamento segnali di INHIBIT

3.5.3 Segnali encoder

Il collegamento degli encoder dei diversi motori non rappresenta una sfida troppo complessa da punto di vista logico: è infatti sufficiente collegare i connettori di tali dispositivi, posizionati sul braccio in prossimità dei motori, percorrere con un cavo la lunghezza dei link fino a scendere nel quadro di destra del Marisa, dove avviene il collegamento con la centralina Beckhoff che saprà gestire i segnali provenienti da questi cavi in modo adeguato, che evidentemente sarà la EL5101. L'aspetto che necessita di un chiarimento, in questa fase, è dato pertanto semplicemente dalla descrizione dei segnali provenienti dalla miriade di cavi provenienti dai singoli encoder, che tra una connessione e l'altra cambiano colore: ci si appresta, quindi, a costruire una tabella "Colore-segnale" per ogni driver.

Occorre riportare inoltre una rappresentazione schematica dei connettori e della nomenclatura usata. Nelle tabelle che seguono i vari segnali e pin Beckhoff fanno riferimento al singolo modulo EL5101 dedicato allo specifico motore.

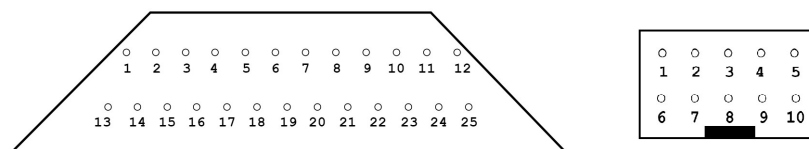


Figura 3.20: Connettori "Struttura" (sinistra) e "Enc M" (destra) usati per i cablaggi degli encoder dei cavi

Abbreviazione	Colore
BK	black
PK	pink
R	red
W	white
Y	yellow
BN	brown
GN	green
GY	grey
BE	blue
PE	purple
O	orange
W.GN	water green

Figura 3.21: Legenda per le abbreviazioni dei colori dei cavi

Nome connettore	Significato
Medio	Connettore su LINK0 lato sx
Enc M	Connettore encoder maschio
Enc F	Connettore encoder femmina
Struttura	Connettore su LINK0 lato dx
BKH IN	Ingressi ai moduli Beckhoff
Segnale	Codice segnale moduli Beckhoff

Figura 3.22: Legenda dei nomi dati ai connettori e locazione nel robot

Medio	Enc M	Struttura	BKH IN	Segnale	Pin BKH
R	1	21	W	5 V	1°
O	7	8	R/BN	0 V	5°
BE	3	20	GY	A	1
GN	8	7	GN	A_NEG	5
W	4	19	BN	B	2
BK	9	6	BE	B_NEG	6

Figura 3.23: Tabella colore-segnale CAVO SX

Medio	Enc M	Struttura	BKH IN	Segnale	Pin BKH
R/W	1	17	GN/BN	5 V	1°
O/BK	7	4	R/BK	0 V	5°
BE/BK	3	16	BK	A	1
GN/BK	8	3	Y	A_NEG	5
W/BK	4	15	Y/R	B	2
BK/W	9	2	W/R	B_NEG	6

Figura 3.24: Tabella colore-segnale CAVO DX

Medio	Enc M	Struttura	BKH IN	Segnale	Pin BKH
PK	1	25	PK	5 V	1°
W	7	12	PE	0 V	5°
Y	3	11	W/BE	A	1
GY	8	24	O	A_NEG	5
GN	4	10	Y/GN	B	2
BN	9	23	W.GN	B_NEG	6

Figura 3.25: Tabella colore-segnale CAVO BACK

Medio	Enc M	Struttura	BKH IN	Segnale	Pin BKH	Enc F
BN	4	2	BE	V	1°	BN
W	7	5	R/BN	0 V	5°	W
GN	5	3	R/BN	A	1	GN
PK	8	6	W/BE	A_NEG	5	R
Y	6	4	PE	B	2	Y
GY	3	7	Y/GN	B_NEG	6	BE

Figura 3.26: Tabella colore-segnale LINK1

Medio	Enc M	Struttura	BKH IN	Segnale	Pin BKH	Enc F
BN	4	10	BN	V	1°	BN
W	7	13	W/BN	0 V	5°	W
GN	5	11	BK	A	1	GN
PK	8	14	GN/BN	A_NEG	5	R
Y	6	12	R/BK	B	2	Y
GY	3	15	Y/R	B_NEG	6	BE

Figura 3.27: Tabella colore-segnale LINK2

Capitolo 4

Algoritmi di controllo e event manager

L'esecuzione di un programma di controllo di un manipolatore risulta spesso articolata in diverse fasi che vengono eseguite seguendo precise sequenze oppure al verificarsi di determinati eventi. Anche per MariBot 2.0 accade lo stesso, basti pensare alla semplice distinzione richiesta nel controllo quando si passa dallo stadio di apprendimento a quello di terapia; affinché i motori rispondano nella maniera adeguata occorre un controllore che sia capace di coordinarne il funzionamento in modo appropriato.

Volendo vedere il problema del controllo del robot in uno spettro più ampio, cioè partendo dall'accensione del dispositivo fino al suo spegnimento, ci saranno evidentemente tutta una serie di operazione intermedie che devono essere considerate: bisogna garantire al terapeuta la possibilità di posizionare l'arto del paziente all'interno dell'ortesi, a valle dell'acquisizione occorre costruire le leggi orarie per le traiettorie da eseguire in esercizio e così via.

In questo lavoro di tesi, dunque, è stato creato un controllore basato sul concetto di macchina a stati, comandato da dei "flag" che si attivano solo al verificarsi di determinate condizioni e che quindi consentono la sincronizzazione dell'intero sistema.

Nel seguente capitolo, quindi, ci si appresta a descrivere il funzionamento complessivo di MariBot 2.0 in tutte le sue fasi, fornendo alcune soluzioni create per

l'implementazione *Simulink*.

4.1 Logica di funzionamento complessivo

Per com'è stata concepita l'idea che descrive l'intera esecuzione del trattamento terapeutico fornito da MariBot 2.0, è possibile individuare sei fasi principali, eseguite in sequenza una dopo l'altra.

Un passaggio preliminare all'esecuzione del codice ma comunque fondamentale è la definizione di tutti i parametri desiderati per la terapia: caratteristiche anatomiche del paziente, specifiche della traiettoria di esercizio (durata, numero di cicli, ecc.) e numero di punti da acquisire in fase di apprendimento. Quest'operazione è eseguita attraverso la modifica dello *SCRIPT_maribot_2PUNTO0*.

A valle di quest'operazione e del posizionamento del paziente in maniera corretta nei confronti del robot, è possibile avviare l'esecuzione del programma, del quale si riportano in sequenza i diversi stadi di avanzamento:

1. Allineamento

Il primo passo da eseguire è che il manipolatore si porti in una configurazione iniziale, che prevede i cavi completamente riavvolti (senza ortesi) e il braccio seriale allineato.

Per raggiungere questa configurazione in sicurezza, occorre prima di tutto definire attraverso il meccanismo di sgancio se il meccanismo dello Scara opererà in braccio destro o sinistro, ergo se la riabilitazione riguarderà l'arto destro o sinistro del paziente. Una volta fatto ciò occorre posizionare il robot in modo che occupi il semipiano sinistro (rispetto al prolungamento del LINK0) se in configurazione destra o il semipiano destro se in braccio sinistro.

A questo punto è possibile avviare il programma, che per prima cosa alzerà i cavi per evitare che i magneti di questi vadano ad urtare il paziente posto sotto al manipolatore, solo successivamente avverrà la movimentazione dei link fino al raggiungimento di completo allineamento (segnalato dai due

sensori induttivi solidali a ciascun link).

Questa fase è necessaria in quanto permette agli encoder dei motori dei link di riconoscere il proprio zero e, conseguentemente, poterlo utilizzare come fattore correttivo nella misura di posizione che verrà impiegata nel controllo delle fasi successive.

In questo stadio tutti i motori sono azionati e controllati in posizione per eseguire lo spostamento richiesto.

2. Setup ortesi

Una volta che il robot ha raggiunto la sua posizione iniziale e ha memorizzato i parametri di azzeramento encoder è possibile per il terapeuta inserire l'arto del paziente nell'ortesi e successivamente agganciare la stessa alle apposite calamite agli estremi dei tre cavi.

Per raggiungere questo obiettivo serve che i link siano liberi di muoversi, pertanto i motori da essi movimentati erogheranno una coppia nulla, mentre i cavi dovranno essere controllati in coppia per garantire il mantenimento delle calamite al di sopra del soggetto.

3. Acquisizione

A questo punto è tutto pronto per l'esercizio: il paziente è correttamente posizionato sotto al robot e l'arto verso cui è mirata la riabilitazione è sostenuto dall'ortesi.

Il terapeuta può ora dare inizio alla terapia vera e propria, che come definito in precedenza, si articola in fase di apprendimento e acquisizione, attraverso la prima pressione del pulsante di acquisizione.

In questo momento i motori dei cavi dovranno essere necessariamente in grado di sostenere il peso del braccio del paziente (controllo di coppia) mentre i link potranno muoversi liberamente secondo quanto deciso dal terapeuta. Queste condizioni fanno sì che il robot si possa muovere all'interno del campo di lavoro specifico per la configurazione di braccio stabilita in precedenza, consentendo quindi la memorizzazione dei punti di passaggio

desiderati per l'esercizio terapeutico.

Il terapeuta può allora movimentare l'arto del paziente e, premendo il pulsante verde del telecomando, memorizzare un certo numero di punti di passaggio desiderati, stabilito durante la configurazione iniziale.

4. Pianificazione traiettoria

Sulla base dei parametri definiti durante la configurazione dei dati di esercizio e dei punti di passaggio acquisiti, si va a implementare la legge di moto che costituirà il riferimento per il controllo in posizione usato in fase di terapia.

In questa fase il controllo dei motori è identico a quello dello stadio di acquisizione: il robot rimane in attesa del via libera per cominciare l'esercizio di riabilitazione.

5. Terapia

Nel momento in cui il terapeuta ritiene più opportuno, alla prima pressione del pulsante successiva a quelle usate per l'acquisizione si ha l'avvio della terapia vera e propria, intesa come il compimento del movimento pianificato in apprendimento del braccio del paziente sfruttando l'assistenza del robot. Evidentemente a questo punto subentra il controllo di posizione di cavi e link: l'implementazione fornisce la possibilità di scegliere tra il classico controllore proporzionale-integrale-derivativo e un controllo adattativo basato su forgetting factor. Per una descrizione più accurata del controllo in terapia, dell'implementazione *Simulink* dei controllori e del confronto tra gli stessi si rimanda ai capitoli 5 e 6.

6. Fine terapia

La terapia si può considerare conclusa o quando effettivamente si raggiunge il termine della traiettoria di riferimento desiderata, cioè qualora si completino tutti i cicli di movimento imposti (Fase IA), oppure nel caso in cui il fisioterapista decida di terminare anticipatamente l'esecuzione attraverso la pressione del pulsante del telecomando (Fase IB).

Essa risulta comunque sempre articolata in due parti. Nel primo stadio (Fase I) si ha l'interruzione del movimento che porta al mantenimento in coppia dell'arto del paziente per quanto riguarda la gestione dei cavi, mentre i link saranno lasciati liberi: questa fase è intesa come il momento in cui il terapeuta può iniziare a subentrare al controllo del robot sostenendo l'ortesi e l'arto del paziente. A questo punto, tramite l'ultima pressione del pulsante verde, si ha anche la "disabilitazione" dei motori dei cavi, con conseguente discesa dell'arto del paziente sorretto dal fisioterapista e possibilità di rimozione dall'end effector (Fase II).

4.2 Macchina a stati

Per riuscire a sincronizzare correttamente il controllo richiesto di volta in volta da ciascuna delle singole fasi di esecuzione del programma, l'approccio utilizzato è quello della macchina a stati o event manager.

Per ogni motore, quindi, è stata definita una variabile "STATO" che modifica il suo valore al verificarsi di determinate condizioni, quali possono essere per esempio la pressione del pulsante di acquisizione per un certo numero di volte oppure la lettura dei sensori induttivi. Il verificarsi di questi diversi eventi vanno ad cambiare il valore di determinati flag che conseguentemente modificano lo STATO di quello specifico motore: a precisi valori di tale variabile corrispondono infine specifici approcci di controllo della tensione di scrittura ai moduli Beckhoff.

Evidentemente quindi, è possibile distinguere solo due tipologie di macchine a stati: una dedicata ai link e una dedicata ai cavi, che poi agiranno nello stesso modo per LINK1 e LINK2 e poi per CAVO SX, CAVO DX e CAVO BACK. Di seguito si riporta la corrispondenza tra fase di esecuzione e relativo stato ad essa associato:

- **Allineamento:** STATO_LINK 0, STATO_CAVI 0
- **Setup ortesi:** STATO_LINK 0, STATO_CAVI 1

- **Acquisizione:** STATO_LINK 1, STATO_CAVI 2
- **Pianificazione traiettoria:** STATO_LINK 1, STATO_CAVI 2
- **Terapia:** STATO_LINK 2, STATO_CAVI 3

- **Fine terapia:**

in questo caso occorre distinguere le due fasi descritte pocanzi:

Fase I: STATO_LINK 4, STATO_CAVI 4

Fase II: STATO_LINK 5, STATO_CAVI 5

A titolo di esempio si riporta lo schema *Simulink* dell'event manager di LINK1 e CAVO SX [Figura 4.1], attraverso cui è possibile comprendere la logica di controllo osservando la distinzione tra le diverse tipologie di ingresso e le uscite.

Gli ingressi si possono infatti sempre dividere in due gruppi: i "flag" e i "write". Sono le variabili che, come descritto in precedenza, modificano il loro valore quando si verificano determinati eventi mentre gli ultimi sono le tensioni in scrittura ai moduli Beckhoff per gli diversi stadi di esecuzione.

In seguito ci si appresterà a descrivere più nel dettaglio questi singoli aspetti del controllo degli stati.

4.2.1 Segnali usati per la sincronizzazione

La commutazione da una fase all'altra dell'esecuzione, con conseguente modifica della tensione "write" in scrittura ai diversi moduli EL4234, è dettata dal cambiamento tra un valore e l'altro dei diversi flag usati durante l'esecuzione del programma che avviene come risposta alla variazione di segnali di diversa natura. Per comprendere meglio come avvenga la gestione della sequenza di operazioni comandate dai diversi flag, si rimanda al diagramma di flusso di Figura 4.4. Volendo descrivere quali sono le fonti di questi "segnali di sincronizzazione", è possibile individuarne tre che di fatto governano l'intera logica dell'event manager e sono:

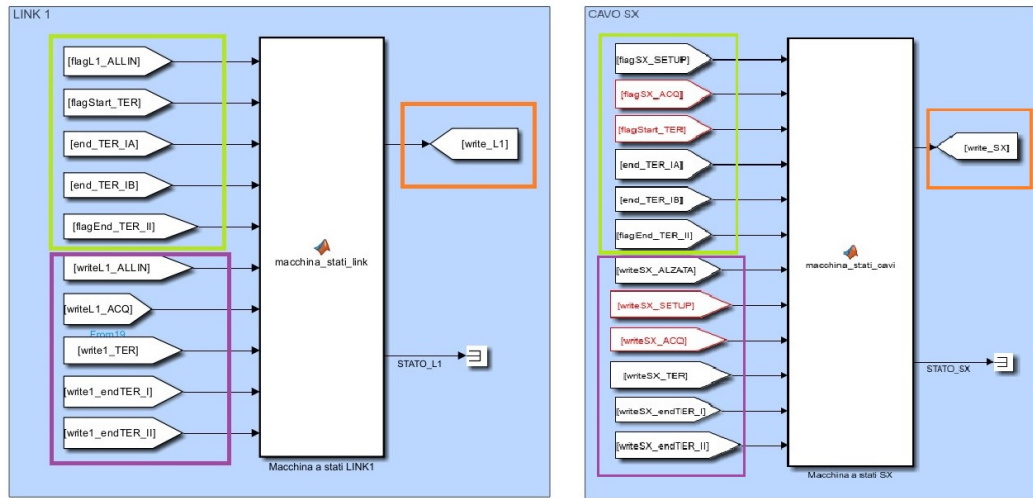


Figura 4.1: Implementazione macchina a stati link e cavi:

verde = flag, viola = write, arancio = scrittura

- **Segnali temporali**

Alcune sequenze del programma di esecuzione vengono gestite in base a segnali temporali dettati da timer che vengono attivati al verificarsi di un determinato evento e al loro scadere determinano il passaggio a uno stadio successivo. Questo tipo di gestione è stata usata, per esempio, nel passaggio da alzata dei cavi ad allineamento dei link: tale fase, infatti, può iniziare solo quando è trascorso il tempo di percorrenza della traiettoria di alzata più un secondo di assestamento.

Gestione analoga si ha per lo stadio di fine terapia, nel caso di fine terapia a ciclo completato: in queste condizioni occorre aspettare due secondi oltre la fine del tempo di ciclo complessivo.

Per questa specifica funzione, in particolare, si sfrutta anche il conteggio del tempo di clock, cioè del tempo totale di simulazione dall'avvio. All'interno del codice di controllo di MariBot 2.0 risulta estremamente utile sia dal punto di vista dell'event manager ma anche, come si vedrà più avanti in questo capitolo, per la pianificazione di traiettoria.

Per riuscire a soddisfare queste funzioni, però, è necessario avere un con-

teggio del tempo che non inizia dall'istante di avvio, ma al raggiungimento dello stadio di inizio terapia: flag "*flagStart_TER*" alto. Il nuovo clock ottenuto è definito "*delayed_clk*" è ottenuto come riportato in Figura 4.2.

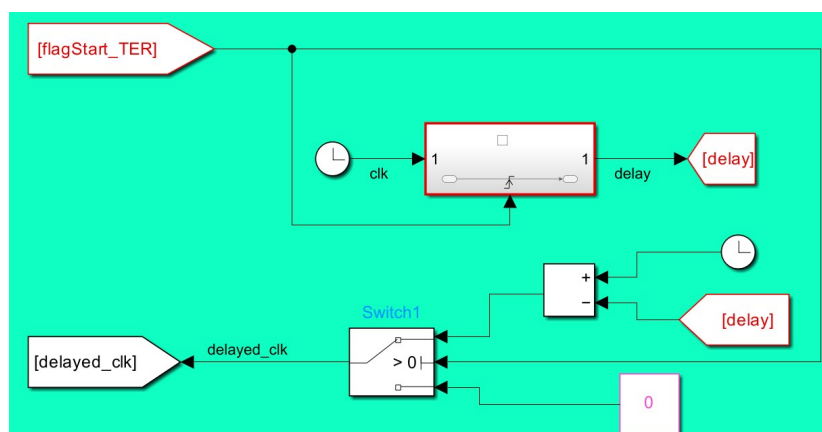


Figura 4.2: Implementazione calcolo "*delayed_clk*"

- **Segnali provenienti dai sensori induttivi**

Il raggiungimento della posizione di allineamento e conseguente memorizzazione della posizione di zero encoder è segnato dalla lettura dei sensori induttivi solidali al LINK0 e al LINK1. Ciascuno di essi comanda un determinato flag di "avvenuto allineamento link" e permette il passaggio allo stadio successivo.

- **Segnali provenienti dal pulsante di acquisizione**

Com'è intuibile dalla descrizione riportata in precedenza, il pulsante di acquisizione rappresenta l'unico dispositivo a portata di mano del terapista durante l'esercizio e pertanto si ha la necessità che questo agisca in modo diverso in punti diversi del programma, fornendo di volta in volta la risposta adeguata e conseguentemente modificando il valore solo dei flag che lo richiedano in quel momento. A tal scopo è stata creata una sezione dello schema di controllo che permette di gestire nel modo corretto l'unico segnale proveniente da questo pulsante.

Occorre quindi distinguere la pressione che abilita la messa in coppia dei

cavi a mantenimento dell'ortesi in fase di apprendimento dalle pressioni richieste per la memorizzazione dei punti di passaggio e infine quelle utili per terminare l'esercizio.

Una questione fondamentale da tenere in considerazione prima di addentrarsi nella parte software della descrizione riguarda l'aspetto fisico del componente utilizzato: il pulsante è monostabile, il che significa che questo garantisce un segnale di tensione alto solamente quando è premuto, mentre cala al terminare della pressione: questo permette quindi sempre di distinguere ogni pressione perchè si commuta da un valore di tensione basso a uno alto.

Consci di questo aspetto, si può ora procedere nell'analisi del blocco di *Gestione del pulsante di acquisizione* [Figura 4.3] e del modello che ha come obiettivo di contare di volta in volta i segnali mandati dal pulsante di acquisizione e memorizzarli nella variabile "*n_imp*" che potrà quindi essere utilizzata nelle diverse parti dello schema di controllo.

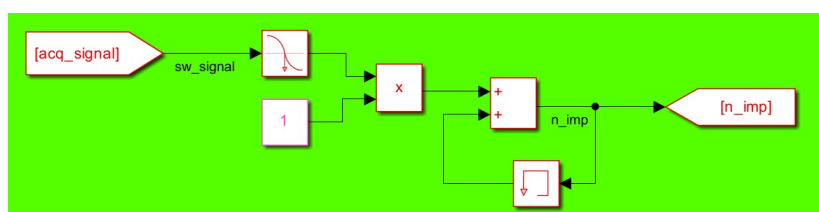


Figura 4.3: Implementazione contatore delle pressioni del pulsante di acquisizione "*n_imp*"

4.2.2 Flag, write e diagramma di flusso

Per descrivere in maniera chiara ed efficace il funzionamento dell'intera macchina a stati si può sfruttare la rappresentazione della sequenza di operazioni mediante un diagramma di flusso.

A tal fine occorre conoscere il significato dei diversi flag e write e in quale fase questi agiscano durante l'esecuzione del programma.

Per brevità si riporta la descrizione dei flag e write che gestiscono l'event manager

dei soli LINK1 e CAVO SX, già visti in precedenza.

Di seguito i flag:

- **flagL1_ALLIN**

Commuta a 1 (allineamento eseguito) quando avviene la lettura del sensore induttivo 1, solidale a LINK0.

- **flagStart_TER**

Commuta a 1 per segnalare l'abilitazione ad avviare la terapia la quale è determinata dalla pressione n_pti+2 (per esempio se si desidera effettuare l'apprendimento di tre punti, l'inizio terapia sarà determinato dalla quinta pressione).

- **end_TER_IA**

Commuta a 1 quando si ha la fine del ciclo di terapia completo, nel dettaglio il flag si alza quando è trascorso il tempo complessivo di moto più ulteriore due secondi di attesa. Ci si porta nella condizione II di fine terapia, ovvero quella del mantenimento del braccio del paziente. L'implementazione del controllore per il mantenimento è denominata "Tensionatore 3".

- **end_TER_IB**

Commuta a 1 quando il terapeuta desidera terminare l'esecuzione prima che sia stato terminato il numero di cicli imposto. La variazione del flag è determinata dalla pressione numero n_pti+3 del pulsante di acquisizione. Anche in questo caso ci si porta allo stadio II di fine terapia.

- **flagEnd_TER_II**

Commuta a 1 quando il terapeuta è pronto a sostenere l'arto del paziente alla fine dell'esercizio. In questa condizione si può arrivare da IA o da IB, che sono raggiunte rispettivamente da segnale temporale oppure da pressione del pulsante di acquisizione. Il flagEnd_TER_II, quindi, deve modificare il

suo valore in due situazioni: se si arriva da IA, allora alla pressione successiva del pulsante (numero n_pti+3) mentre se si arriva da IB occorre tener in considerazione la pressione aggiuntiva che ha abilitato il flag `end_TER_IB`, pertanto la variazione sarà data dalla pressione numero n_pti+4 .

- **flagSX_SETUP**

Commuta a 1 quando sono state completate le procedure di raggiungimento della configurazione iniziale, dunque di alzata dei cavi e di allineamento dei link, cui consegue lo stadio di setup con posizionamento del braccio del paziente nell'ortesi. La commutazione avviene quando entrambi i membri hanno raggiunto la loro posizione di zero, cioè quando entrambi i flag `flagL1_ALLIN` e `flagL2_ALLIN` valgono 1. In questa fase si mantengono i cavi (senza ortesi) in controllo di coppia. Il controllore per il mantenimento è denominato "Tensionatore 1".

- **flagSX_ACQ**

Commuta a 1 per abilitare il mantenimento del braccio del paziente per rendere il robot pronto alla successiva fase di apprendimento. Questa fase è abilitata attraverso la prima pressione del pulsante di acquisizione. IN questo caso si utilizzerà un diverso controllore per il mantenimento, detto "Tensionatore 2".

Si riportano adesso i diversi write. Questi rappresentano le diverse uscite che i differenti controllori implementati nello schema vorrebbero andare a fornire ai moduli di scrittura Beckhoff. Evidentemente a ogni fase sarà necessario far arrivare a tali moduli solo la tensione d'uscita corretta (`write_L1` e `write_SX`), generata dal controllore di quello specifico stadio del programma.

Si possono individuare quattro tipi di write in tutto il codice di controllo: i write nulli (motori non azionati), i write di movimento semplice, quelli di mantenimento (i cosiddetti "Tensionatori") e il movimento terapeutico, in cui la tensione d'uscita è generata dal controllore adattativo.

Di seguito si riportano più nel dettaglio i diversi segnali di write per LINK1 e CAVO SX.

- **writeL1_ALLIN**

Tensione di scrittura in allineamento per LINK1: data da controllo di posizione con PID che spazza il campo di lavoro fino al raggiungimento del sensore induttivo.

- **writeL1_ACQ**

Tensione di scrittura in setup e acquisizione per LINK1: motore non azionato, può essere movimentato liberamente dal terapeuta.

- **writeL1_TER**

Tensione di scrittura in terapia per LINK1: data dal controllore adattativo (vedere nel dettaglio capitolo 5).

- **writeL1_endTER_I**

Tensione di fine terapia fase I per LINK1: motore non azionato, può essere movimentato liberamente dal terapeuta.

- **writeL1_endTER_II**

Tensione di fine terapia fase II per LINK1: motore non azionato, può essere movimentato liberamente dal terapeuta.

- **writeSX_ALZATA**

Tensione di alzata per CAVO SX: data da controllo di posizione con PID che alza i soli cavi (più magneti) fino a quota desiderata decisa da script (in alto per evitare di urtare il paziente).

- **writeSX_SETUP**

Tensione di setup per CAVO SX: data da "Tensionatore 1" che garantisce il mantenimento in coppia dei soli magneti di sostegno dell'ortesi. In questo modo il fisioterapista può posizionare correttamente l'arto del paziente e, sostenendolo, connetterlo ai cavi del robot tramite l'ortesi.

- **writeSX_ACQ**

Tensione di acquisizione per CAVO SX: data da "Tensionatore 2" che garantisce il mantenimento in coppia del braccio del paziente.

- **writeSX_TER**

Tensione di setup per CAVO SX: data dal controllore adattativo (vedere nel dettaglio capitolo 5).

- **writeSX_endTER_I**

Tensione di fine terapia fase I per CAVO SX: data da "Tensionatore 3" che garantisce il mantenimento in coppia del braccio del paziente. In questo modo il terapeuta può sostenere l'arto e, alla pressione del pulsante di acquisizione, passare alla fase II.

- **writeSX_endTER_II**

Tensione di fine terapia fase II per CAVO SX: motori dei cavi non azionati.

Si riporta di seguito il diagramma di flusso del funzionamento complessivo del manipolatore.

Lo schema, per risultare più facilmente interpretabile, riporta solo i flag che variano da una fase all'altra dell'esecuzione; i flag impiegati nella rappresentazione fanno riferimento a un unico, generico, "CAVO" in quanto la logica di gestione è del tutto analoga tra CAVO SC, DX e BACK; per i link si distinguono invece i flag di LINK1 e LINK2.

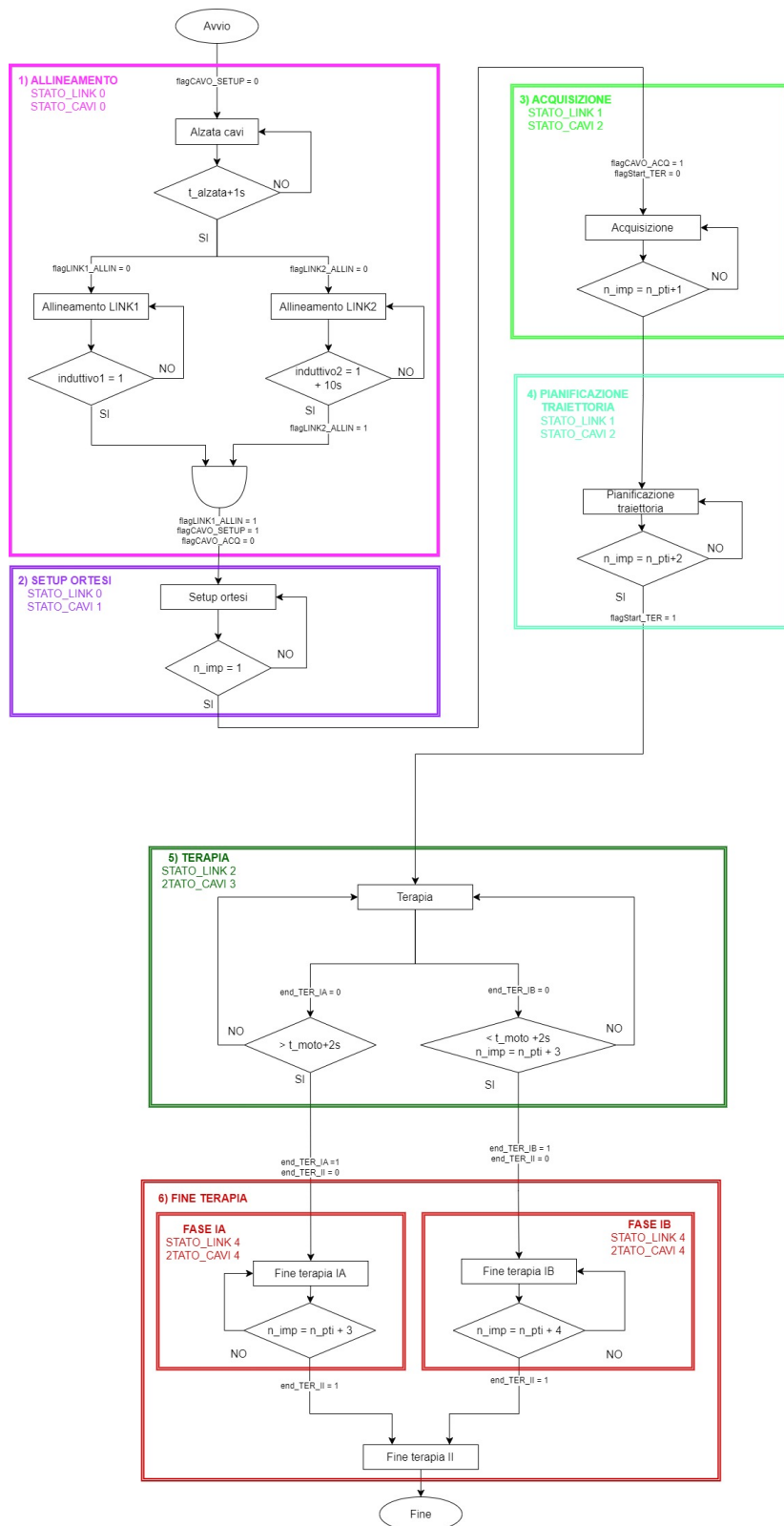


Figura 4.4: Diagramma di flusso del funzionamento complessivo

4.3 Implementazione degli stati

Il problema della sincronizzazione degli eventi rappresenta solo uno degli aspetti da gestire per riuscire a controllare in maniera adeguata MariBot 2.0. Occorre infatti implementare degli schemi di controllo corretti per ciascuno degli stadi dell'esecuzione del programma visti in precedenza.

In alcuni casi l'implementazione è pressochè identica per diverse fasi, basti pensare per esempio ai tensionatori usati per il mantenimento dei cavi in coppia; risulta quindi opportuno portare un'unica descrizione generale dello schema implementato per questo tipo di controllo.

4.3.1 Controllo in posizione

Per la movimentazione di cavi e link al di fuori del trattamento terapeutico vero e proprio si utilizza in tutti i casi un controllo in posizione basato su controllore PID.

Questo tipo di controllo richiede la creazione di una traiettoria di riferimento la quale, confrontata con le misure di posizione fornite dagli encoder, permette di determinare un errore che rappresenta l'ingresso al controllore. Questo, poi, attraverso una corretta dei guadagni proporzionale, integrale e derivativo, consente un corretto inseguimento del riferimento.

Un controllo di questo tipo è stato utilizzato per l'intera fase di allineamento, quindi di alzata dei cavi e di rotazione dei link fino allo zero.

In quest'ultimo caso, in particolare, si usano in abbinata due PID: il primo è quello che esegue il movimento di percorrenza dello spazio di lavoro, con traiettoria di riferimento generata dallo script in base alla configurazione di braccio destro o sinistro selezionata; una volta raggiunta la posizione di allineamento (in corrispondenza della lettura da parte del sensore induttivo) si memorizza tale valore e lo si usa come riferimento per il secondo controllore. Questo svolge una funzione di mantenimento della posizione di zero del LINK1 fino al termine dell'allineamento del LINK2.

Si riporta in Figura 4.5 lo schema per il controllo dell'allineamento del LINK1

(LINK2 si comporta allo stesso modo).

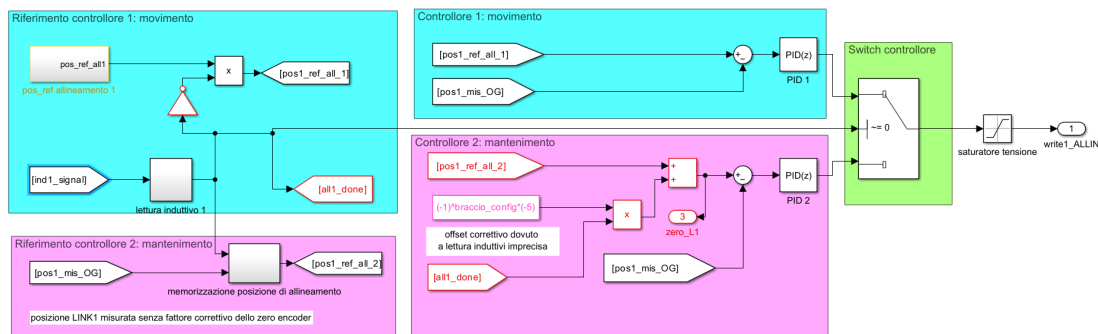


Figura 4.5: Implementazione del controllo in fase di allineamento per LINK1

4.3.2 Controllo in coppia: tensionatori

Il mantenimento in coppia dei cavi permette che questi possano essere liberamente movimentati dal terapeuta o dal paziente stesso garantendo sempre la possibilità di sostenere completamente il peso dell'arto del soggetto ma al contempo non svolgendosi qualora si rendesse necessario un sollevamento dell'ortesi. L'obiettivo del meccanismo tensionatore è, quindi, garantire che sui cavi sia sempre presente la stessa tensione, in ogni istante e posizione.

L'implementazione realizzata per raggiungere tale scopo sfrutta ancora una volta un controllore PID, questa volta mirato a ridurre l'errore su una forza di riferimento.

Il riferimento, in questo caso, è infatti la coppia di tiro che deve essere mantenuto costante sui cavi per garantire l'equilibrio del sistema braccio/ortesi. Perché ciò si verifichi, serve che il tiro eguagli la forza peso del sistema appena definito, il quale dipende per ovvie ragioni dalla costituzione e dalle caratteristiche anatomiche del paziente.

In Figura 4.6 si può infatti osservare che la coppia di riferimento dipende dalla massa che è necessario sostenere, che di fatto rappresenta l'unica differenza tra Tensionatore 1, 2, e 3. La retroazione è invece garantita dalla forza stimata

partendo dalla misura di corrente, secondo la relazione:

$$C_{mis} = i_{mis} \cdot K_t \cdot i \quad (4.1)$$

dove K_t rappresenta la costante di coppia del motore e i il rapporto di trasmissione. Lo schema di Figura 4.6 riporta l'implementazione utilizzata per Tensionatore 2, dunque addetto al mantenimento dei cavi in fase di acquisizione. Evidentemente la massa da sostenere, al netto di quella dell'ortesi e dei magneti, che rimane invariata, dipende dalla costituzione e da altre caratteristiche anatomiche del soggetto cui è destinata la terapia, pertanto è necessario che il riferimento sia modificabile attraverso l'impostazione dei parametri in fase di configurazione iniziale. Per garantire il mantenimento, inoltre, si è ipotizzato che ciascuno dei tre cavi debba sorreggere un terzo della forza peso totale esercitata.

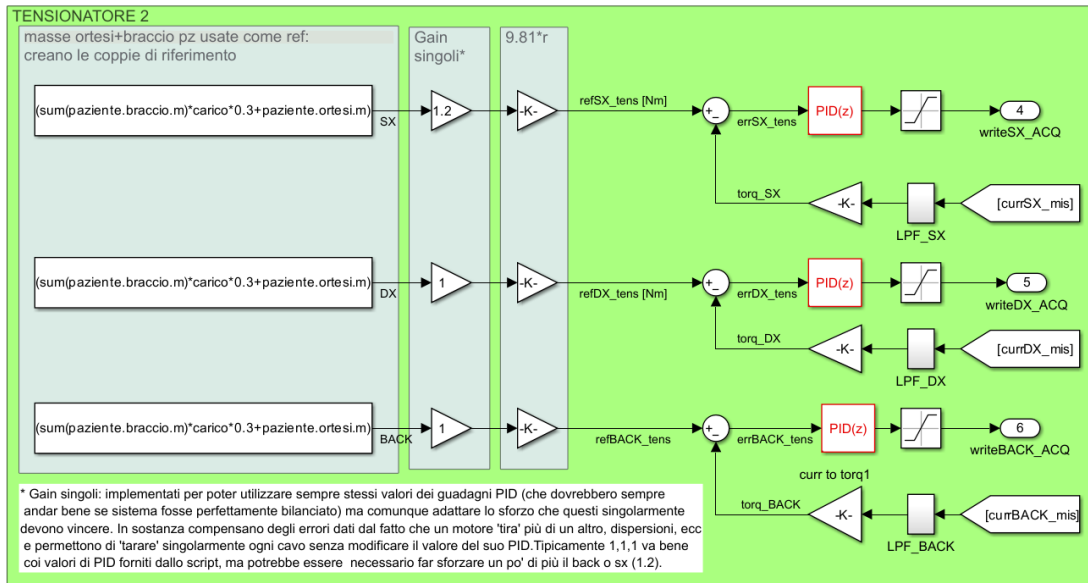


Figura 4.6: Implementazione del controllo di coppia per Tensionatore 2

4.3.3 Acquisizione

La sezione dello schema dedicata all'apprendimento fornisce la possibilità di memorizzare un certo numero di punti di passaggio desiderati per la generazione della traiettoria di esercizio. Questi punti, il cui numero n_{pti} è un parametro scelto in fase di configurazione iniziale, sono acquisiti attraverso la pressione del

pulsante verde del telecomando e sono utilizzati in seguito l'interpolazione che porta alla creazione della traiettoria di riferimento di posizione per il controllore adattativo in fase di terapia.

Nell'implementazione suggerita, la procedura di acquisizione termina con la creazione di cinque vettori (uno per motore) nel quale sono salvati le posizioni lette dagli encoder in corrispondenza del fronte di discesa del segnale proveniente dal pulsante. Affinchè il passaggio al pianificatore risulti più immediato, si è deciso di imporre che la dimensione di questo vettore (`posSX_ACQ`, `posDX_ACQ` e così via) corrisponda sempre al massimo numero di punti memorizzabili, cioè cinque. Nel caso in cui si decidesse di acquisire un numero minore di punti (per esempio se $n_pti=3$) il vettore di acquisizione vedrà i suoi ultimi due elementi posti a zero e non saranno considerati per l'interpolazione.

Riportando, per esempio, lo schema realizzato per l'acquisizione delle posizioni di passaggio per il CAVO SX (Figura 4.7) si distinguono tre blocchi logici:

1. Distinzione in base al numero di punti da acquisire

In base al valore di n_pti si abilita solo uno tra i successivi blocchi di memorizzazione dei punti.

2. Creazione vettore dei soli punti acquisiti

In base al segnale di enable generato da 1) si crea un vettore "parziale" di acquisizione (`posSX_2pti`, `posSX_3pti`, ecc.) contenente solamente le posizioni acquisite attraverso il pulsante.

3. Creazione vettore finale di acquisizione

Spostamento del vettore parziale creato in 2) all'interno del vettore "definitivo" da usare per la pianificazione, costituito da cinque elementi.

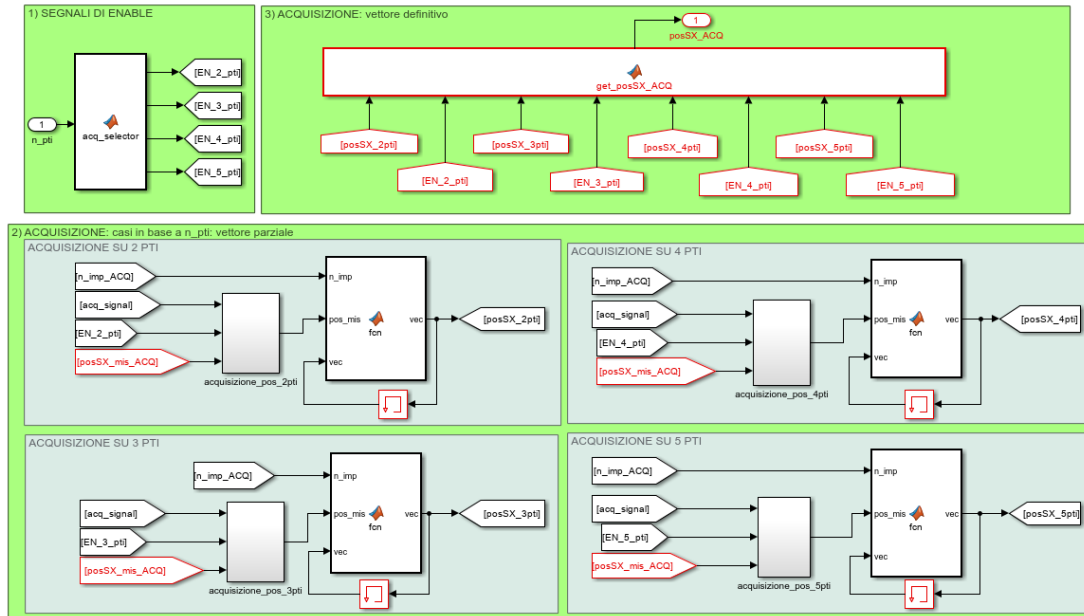


Figura 4.7: Implementazione della parte di acquisizione e apprendimento

4.3.4 Pianificazione traiettoria

La parte di pianificazione di traiettoria è quella sezione dello schema di controllo addetta alla generazione della traiettoria di riferimento per interpolazione dei punti memorizzati in fase di acquisizione.

Per arrivare a questo obiettivo è stato necessario affrontare due problemi, primo dei quali la scelta della legge di moto più adeguata durante l'esercizio e la sua suddivisione in fasi. Il secondo aspetto da affrontare è stato la questione dell'implementazione *Simulink* del pianificatore.

1. Legge di moto

Uno dei vantaggi dell'impiego di robot nel processo di riabilitazione è, com'è noto, quello di riuscire a garantire assistenza continuativa al paziente per lunghe sedute di terapia senza necessariamente richiedere il costante contributo del terapeuta. Si rende quindi necessario implementare leggi che possano potenzialmente essere ripetute in maniera ciclica, per un numero di volte definito in fase di configurazione iniziale.

- **Definizione fasi della ciclica**

Tra gli altri parametri che occorre impostare rientrano poi le durate delle diverse fasi del ciclo. Per MariBot 2.0 si garantisce sempre la possibilità di suddividere un singolo ciclo di esercizio nelle quattro fasi di: attesa iniziale, andata, fermo intermedio e ritorno. Sarà in ogni caso possibile modificare il valore di queste durate e, per esempio, eliminare le pause iniziali tra un ciclo e l'altro.

Un ultimo accorgimento che è stato implementato riguarda l'inserimento della fase di "setup", intesa come la movimentazione iniziale che porta il braccio del paziente alla posizione di inizio ciclo e dell'"attesa finale", con mantenimento della posizione raggiunta a fine esercizio.

Si può quindi rappresentare la durata complessiva del tempo di terapia come:

$$t_{tot} = t_{setup} + n_{cicli} \cdot (t_{att_iniz} + t_{andata} + t_{fermo_int} + t_{ritorno}) + t_{att_fin} \quad (4.2)$$

- **Definizione leggi di moto**

La scelta delle leggi di moto più adatte nelle diverse fasi che costituiscono l'esercizio può essere semplificata facendo alcune considerazioni preventive.

Prima di tutto, risulta ovvio che negli stadi di fermo (iniziale, intermedio o finale) la traiettoria risulterà in un andamento in posizione costante nel tempo, con velocità e accelerazioni nulle.

In secondo luogo è utile osservare come la fase di setup possa essere interpretata come una legge punto-punto tra l'ultima posizione assunta dai diversi motori alla pressione del pulsante corrispondente all'inizio della terapia e il primo punto tra quelli acquisiti.

A tal fine sono state valutate diverse leggi di moto per questa fase: trapezoidali in velocità, cubiche e polinomiali. Per garantire un elevato livello di dolcezza e la continuità fino al secondo grado della derivata in tutti i suoi punti, per la traiettoria di riferimento di posizione di setup si è deciso di implementare una polinomiale di quinto grado.

L'esecuzione ciclica, d'altro canto, non può essere interpretata in una

semplice legge punto-punto, in quanto essa deve utilizzare necessariamente tutti i punti memorizzati in fase di apprendimento, da essi effettuare un'interpolazione e così ottenere la legge di moto cercata. Per raggiungere tale obiettivo è stato utilizzato il metodo della S.P.Line¹ (di seguito semplicemente "spline").

Questa consente di ottenere un'interpolazione su N punti attraverso $N-1$ polinomi di grado ridotto i quali, nella sua implementazione di base, sono polinomi di terzo grado. Questo tipo di spline, tipicamente chiamata spline cubica o 3-3-3, permette sempre il passaggio per i punti di interpolazione desiderati in precisi istanti di tempo fissati, garantisce poi continuità di velocità e accelerazione nei punti intermedi e fornisce la possibilità di imporre i valori di velocità agli estremi.

L'impiego di una spline 3-3-3 non risolve eventuali problematiche di discontinuità in accelerazione, soprattutto nei passaggi da fasi di fermo a fasi di moto. Tale aspetto risulta comunque poco influente nella dinamica del manipolatore, anche grazie alla sua elevata rigidità strutturale, che permette di poter approssimare la dinamica di accelerazione con quella di corrente, che permette l'inseguimento del gradino di accelerazione con un transitorio ridotto.

Si può quindi rappresentare matematicamente l'intera legge di moto come riportato nelle 4.3, 4.4, 4.5, 4.6, 4.7 e 4.8, la cui rappresentazione grafica è invece fornita in Figura 4.8.

Con:

$$j = 1, 2, 3, \dots, n_cicli;$$

$$i = 1, 2, 3, \dots, n_pti-1;$$

$$q_{in} = \text{posizione iniziale};$$

$$q_{fin} = \text{posizione finale};$$

SETUP: poli5

$$t \in [0, t_{setup}-M(1)]$$

¹Smooth Path Line

$$q(t) = a_5 t^5 + a_4 t^4 + a_3 t^3 + a_2 t^2 + a_1 t + a_0 \quad (4.3)$$

ATTESA INIZIALE: costante

$$t \in [t_{setup_M}(j), t_{att_iniz_M}(j)]$$

$$q(t) = q_{in} \quad (4.4)$$

ANDATA: spline cubica

$$t \in [t_{att_iniz_M}(j), t_{andata_M}(j)]$$

$$q(t) = a_{0,i,j} + a_{1,i,j}(t - t_{i,j}) + a_{2,i,j}(t - t_{i,j})^2 + a_{3,i,j}(t - t_{i,j})^3 \quad (4.5)$$

FERMO INTERMEDIO: costante

$$t \in [t_{andata_M}(j), t_{fermo_int_M}(j)]$$

$$q(t) = q_{fin} \quad (4.6)$$

RITORNO: spline cubica

$$t \in [t_{fermo_int_M}(j), t_{ritorno_M}(j)]$$

$$q(t) = a_{0,i,j} + a_{1,i,j}(t - t_{i,j}) + a_{2,i,j}(t - t_{i,j})^2 + a_{3,i,j}(t - t_{i,j})^3 \quad (4.7)$$

ATTESA FINALE: costante

$$t \in [t_{ritorno_M}(j), t_{attesa_fin}]$$

$$q(t) = q_{in} \quad (4.8)$$

2. Implementazione

L'implementazione (riportata per intero in Appendice A) che permette la creazione della traiettoria pianificata e descritta nella sezione precedente, comincia con la lettura dei punti memorizzati nel vettore a cinque elementi e dalle posizioni misurate alla pressione del pulsante di inizio terapia, denominate di STARTER. Da questi dati occorre costruire la legge per la fase di setup e per l'esercizio. Se è semplice costruire la traiettoria di setup iniziale, perchè di tipo punto-punto, lo stesso non si può dire per la gestione della

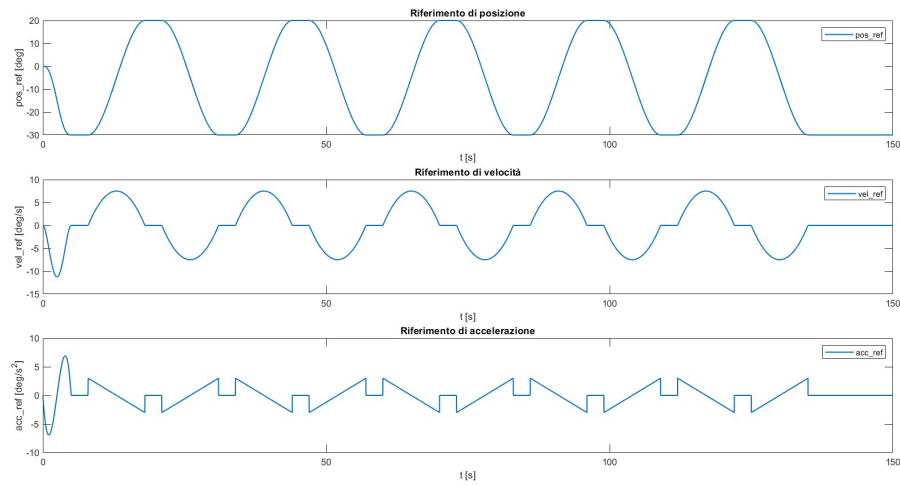


Figura 4.8: Leggi di moto di riferimento con: $n_cicli = 5$; $t_fermo_intermedio = t_attesa_iniziale = t_attesa_finale = 3s$; $t_andata = t_ritorno = 10s$

ciclica.

I maggiori problemi rilevati infatti risiedono nella natura intrinseca del funzionamento software di *Simulink* il quale, con un periodo di campionamento definito di un millesimo di secondo, analizza e ricompila l'intero codice. Questo si ripercuote nell'impossibilità di creare la traiettoria ciclica semplicemente replicando un singolo ciclo per il numero n_cicli di volte sfruttando un *for loop*, in quanto il software va, ad ogni esecuzione, a fornire solo la prima iterazione del ciclo, rendendo vano questo approccio di gestione.

A tal fine si è adottato come soluzione un approccio basato sul precalcolo degli intervalli di tempo delle diverse fasi del ciclo, per ognuno dei diversi cicli di esercizio impostati.

Questo significa che per ogni fase (es: andata) è costruito un vettore contenente i diversi istanti di tempo finale della fase di andata per ciascun ciclo, che saranno pertanto sfalsati l'uno dal successivo di un periodo completo (es: t_andata_M). Attraverso un puntatore inizializzato a zero (j) si andrà ad accedere, di ciclo in ciclo, ad un specifico "set" di istanti di tempo, inerenti a tutte le fasi del corretto ciclo j -esimo. Questi istanti temporali verranno quindi utilizzati come gli istanti in cui imporre il passaggio per i punti di passaggio acquisiti, risultando nella creazione di traiettoria attra-

verso il comando *spline* di *MATLAB*, nel quale sono inoltre stati imposti i valori di velocità nulli agli estremi. Questo vuol implicitamente significare che tra andata e ritorno il braccio del paziente deve in ogni caso fermarsi. Come già descritto in precedenza, il tempo su cui si basa l'intera esecuzione del pianificatore è il *delayed_{clk}*, questo per garantire una omogeneità nel conteggi del tempo di simulazione e dei riferimenti temporali forniti alla spline.

Capitolo 5

Controllo a basso livello

La fase del controllo in terapia è quella che più interessa ai fini del funzionamento del manipolatore in ambito medico in quanto da essa dipende l'efficacia complessiva del trattamento riabilitativo.

MariBot2.0 si propone di offrire un processo terapeutico basato su controllori adattativi per link e cavi.

Nella sua versione originale, però, il controllo era affidato a semplici controllori PID, capaci di garantire l'esecuzione della traiettoria desiderata ma senza quelle caratteristiche di attitudine alla collaborazione col paziente proprie del controllo adattivo.

Nel seguente capitolo verrà esposta la logica dei controllori proporzionali-integrali-derivativi ("a basso livello") e si riporterà il confronto tra i risultati ottenuti per via simulativa e quelli sperimentali.

5.1 Controllore PID

L'architettura del controllore PID risulta estremamente diffusa in diversi ambiti in quanto fornisce un approccio standard alla gestione del processo osservato.

Noto il suo comportamento, infatti, è sufficiente essere in grado di settare correttamente i parametri che lo caratterizzano per soddisfare definite esigenze o modificare determinate prestazioni per poter controllare nel modo desiderato il

sistema.

Dalla teoria dei controlli automatici è noto che per sistemi continui la funzione di trasferimento che descrive questa particolare logica considera descrive l'uscita $u(t)$ del controllore in funzione dell'errore in ingresso $e(t)$, definito come la differenza tra una grandezza di riferimento $r(t)$ e una opportunamente fornita in retroazione $r'(t)$.

Nel caso del manipolatore in esame, l'errore considerato è calcolato tra la traiettoria di riferimento di posizione e la misura ottenuta dagli encoder.

La funzione di trasferimento nel dominio di Laplace è quindi:

$$C(s) = \frac{U(s)}{E(s)} = K_p + \frac{K_i}{s} + K_d \cdot s \quad (5.1)$$

In questa formulazione si possono individuare tre termini, ciascuno opportunamente scalato per un certo guadagno:

- **Termine proporzionale**, proporzionale secondo K_p (guadagno proporzionale) all'errore sul riferimento, la sua azione permette al controllore di non discostarsi in maniera troppo elevata rispetto al valore desiderato;
- **Termine integrale**, proporzionale secondo K_i (guadagno integrale) all'integrale dell'errore, di fatto quindi dipende dal valore medio dello stesso e cerca di ridurlo, permettendo così di annullare l'errore a regime;
- **Termine derivativo**, proporzionale secondo K_d (guadagno derivativo) alla derivata dell'errore, risente fortemente della velocità di variazione dell'errore

In Figura 5.1 si riporta lo schema generale nel dominio di Laplace per il controllo di un processo $G(s)$.

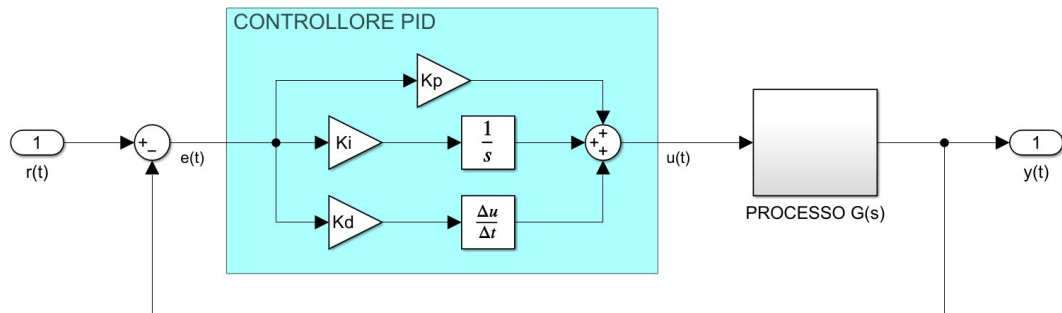


Figura 5.1: Generico schema di controllo in architettura PID

5.2 Simulazioni cavi

In questa sezione ci si propone di presentare il modello dinamico impiegato per la simulazione del comportamento dei cavi del manipolatore. Tale modello tornerà utile sia per la parte inerente al controllo PID ma anche per le simulazioni del sistema con adattativo.

Verranno riportate le prove per la movimentazione ciclica sia del singolo cavo che dei tre cavi con paziente.

5.2.1 Modello del sistema

Il modello impiegato per le simulazioni, riportato in Figura 5.2, è quello di un sistema di sollevamento supposto agente nella sola direzione verticale, quindi descritto da un unico grado di libertà.

Il modello esposto fa riferimento al caso del singolo cavo ma è possibile estendere gli stessi ragionamenti anche per il complessivo dei tre motori, in quanto sarà sufficiente modificare in maniera coerente i parametri meccanici ed elettrici e definire, da simulazione, come la massa totale di ortesi e braccio del paziente sia da considerarsi distribuita in maniera che al singolo cavo BACK sia richiesto il sollevamento che complessivamente cavo SX e DX forniscono, in quanto la conformazione con due cavi davanti e uno dietro suggerisce una non omogenea distribuzione delle forze richieste per il mantenimento.

Il sistema in esame modella i componenti reali ed è perciò costituito da:

- **Motore**, inerzia J_m , rapporto di riduzione interno τ e rendimento η ;
- **Puleggia**, inerzia J_p , raggio r_p ;
- **Filo inestensibile**;
- **Massa**, massa m .

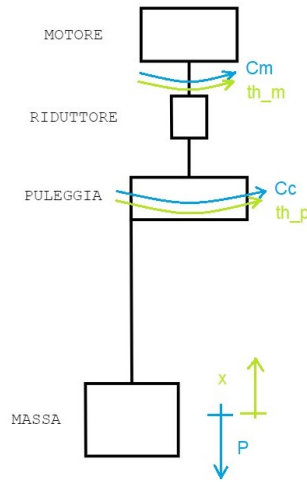


Figura 5.2: Modello del sistema: in azzurro le forze e coppie agenti, in verde gli spostamenti

Il modello dinamico è stato ricavato sfruttando il bilancio delle potenze per un sistema in moto diretto.

Tale teorema eguaglia la sommatoria delle potenze derivanti da forze e coppie esterne con la derivata nel tempo dell'energia cinetica dei diversi corpi, come segue:

$$\frac{dE_c}{dt} = \sum_i W_i \quad (5.2)$$

Si calcolano i due contributi in modo da poter ottenere una formulazione per la coppia motrice richiesta per il sollevamento.

$$\frac{dE_c}{dt} = \frac{d}{dt} \left(\frac{1}{2} J_m \dot{\theta}_m^2 + \frac{1}{2} J_p \dot{\theta}_p^2 + \frac{1}{2} m \dot{x}^2 \right) \quad (5.3)$$

$$\sum_i W_i = C m \dot{\theta}_m - m g \dot{x} \quad (5.4)$$

$$C_m = m g \left(\frac{r_p \cdot \tau}{\eta} \right) + \left(J_m + J_p \left(\frac{\tau^2}{\eta} \right) + m \left(\frac{r_p^2 \cdot \tau^2}{\eta} \right) \right) \ddot{\theta}_m \quad (5.5)$$

Dall'inversione di tale relazione è possibile ottenere l'accelerazione e, integrando due volte, la posizione da utilizzare nel ramo di feedback dello schema.

5.2.2 Risultati simulativi cavi

Di seguito si riportano i risultati ottenuti tramite simulazione con schema *Simulink* del comportamento del controllore PID applicato ai cavi. Ciò che si analizzerà, nel dettaglio, riguarda l'esecuzione della traiettoria ciclica in fase di terapia (definita secondo le leggi definite dal pianificatore nel capitolo 4), dapprima considerando il singolo cavo cui è stata collegata una massa di 0.6 kg e successivamente l'intera ortesi. Come accennato in precedenza, il tipo di ortesi Baita impiegato in MariBot 2.0 vede il collegamento di due cavi nella parte anteriore e uno solo nella parte posteriore della stessa: questo fenomeno evidentemente si ripercuote in una non omogenea distribuzione delle forze, quindi delle coppie motrici richieste per il movimento. Si suppone che lo sforzo richiesto sia uguale per i cavi DX e SX, mentre che il BACK debba garantire da solo la stessa forza sviluppata complessivamente dai due. Ciò che ci si attende dalle simulazioni è, una volta settati correttamente i guadagni del PID, l'inseguimento delle posizione e velocità di riferimento e al contempo l'oscillazione della forza sviluppata dal controllore nell'intorno del peso applicato al singolo cavo, quindi 6N per le simulazioni del singolo cavo, mentre per le prove con tutti i tre cavi ci si aspetta, come descritto pocanzi, $F_{SX} = F_{DX} = 1/3 P$, $F_{BACK} = 2/3 P$ (dove P è la forza peso data dalla massa dell'ortesi).

- **Simulazioni singolo cavo con massa**

Le simulazioni a basso livello hanno richiesto la taratura dei guadagni del controllore: K_p , K_i e K_d . Maggiore interesse in questo lavoro di tesi è stato rivolto all'osservazione del comportamento della componente integrale in quanto essa verrà sostituita con la logica adattativa nel controllo ad alto livello. Effettuando prove basate su procedura di *trial and error* è stato possibile determinare i guadagni proporzionali e integrali: $K_p=1000$,

$K_d = K_p/200$. Si sono poi valutati i diversi comportamenti al variare del K_i . Come si evince dalle figure che seguono, diversi valori di tale parametro portano il sistema alla correzione dell'errore a regime in maniera più rapida e decisa (per tutte le prove svolte è riportato l'andamento dell'errore percentuale "misurato" in trasparenza, mentre in grassetto si riporta la media dello stesso errore, ottenuto come media mobile pesata in un intervallo temporale variabile in funzione della frequenza delle oscillazioni misurate).

Ciò è determinato dal polo, in termini di trasformata di Laplace, introdotto dall'azione integrale stessa che, se posizionato in bassa frequenza, determina una riduzione della costante di tempo per la risposta del sistema (fino a limiti che possono portare, al contrario, a condizioni di instabilità).

Si osserva poi come la forza necessaria per la movimentazione si porti ad oscillare nell'intorno dei 7N. Evidentemente il valore medio ideale dovrebbe essere nell'intorno della massa sollevata, cioè circa 6N, tuttavia il modello proposto nel paragrafo 5.2.1 implementa anche le non idealità rappresentate dai rendimenti dei motori ottenuti da datasheet, che fanno sì che la forza che deve essere effettivamente erogata per lo scopo sia maggiore rispetto al caso ideale.

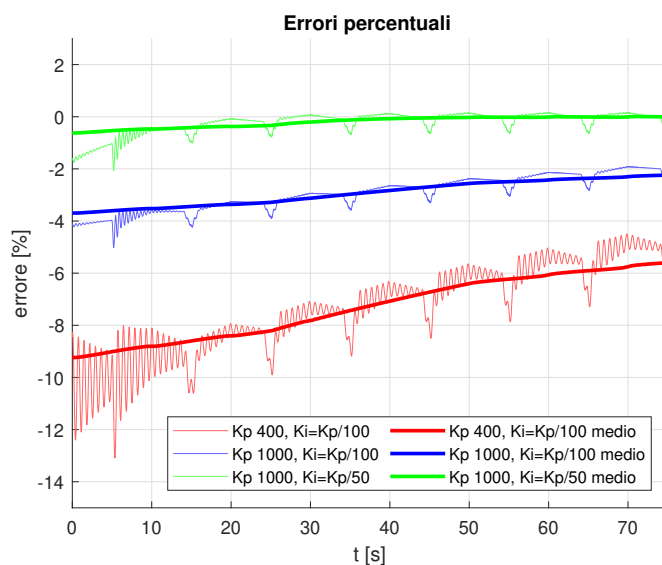


Figura 5.3: Errori di posizione percentuali al variare dei guadagni per singolo cavo

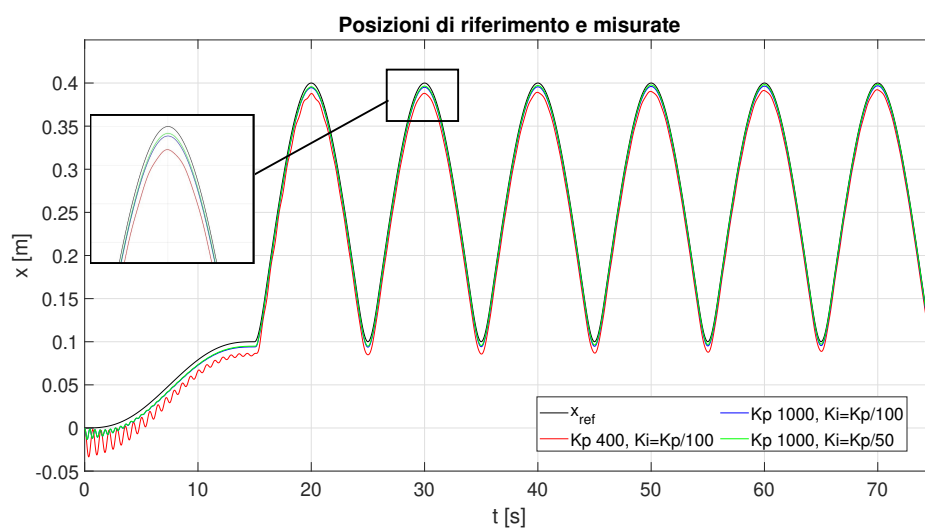


Figura 5.4: Inseguimento di posizione per singolo cavo

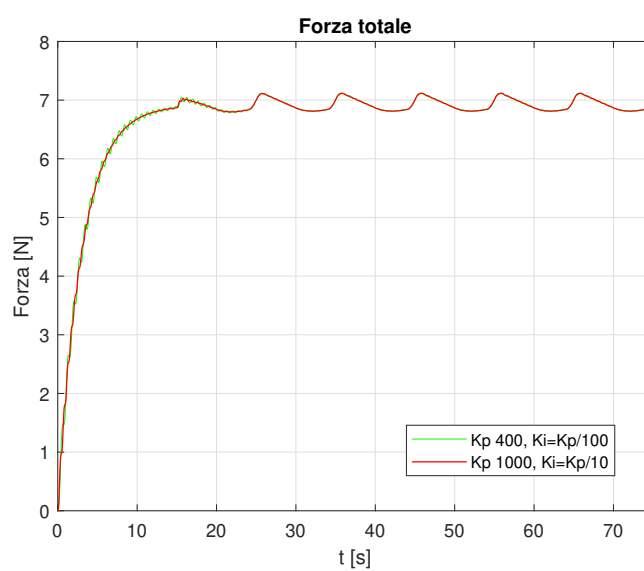


Figura 5.5: Forza risultante per singolo cavo

- **Simulazioni tre cavi con ortesi**

Il procedimento e i metodi da seguire in questo caso sono è analoghi a quello eseguito per il singolo cavo tuttavia, quando si considera il sistema dei tre motori cui si aggiunge l'ortesi, è necessario tenere conto che, come accennato in precedenza, il tipo di end-effector impiegato vede un solo cavo nella parte posteriore e pertanto sarà soggetto a uno sforzo maggiore nel sollevamento in quanto da solo deve fornire la forza che complessivamente erogano i cavi anteriori.

Ciò si ripercuote direttamente sulla scelta dei guadagni del controllore.

In prima battuta, infatti, si è pensato di strutturare il controllo in modo che i PID per i tre cavi fossero settati nello stesso modo, quindi con guadagni identici. Proprio per il motivo esposto pocanzi, questo approccio porta il cavo BACK ad avere un errore sull'inseguimento del riferimento maggiore rispetto ai cavi SX e DX per i quali, invece, risulta uguale.

Per correggere tale aspetto, dunque, si rende necessario fornire un K_p maggiore al cavo posteriore in modo che porti alla medesima correzione sull'inseguimento fornita dai cavi anteriori.

In Figura 5.6 sono riportati gli andamenti dell'errore percentuale sul riferimento nel caso di taratura con guadagni identici e nel caso di taratura differenziata per il BACK.

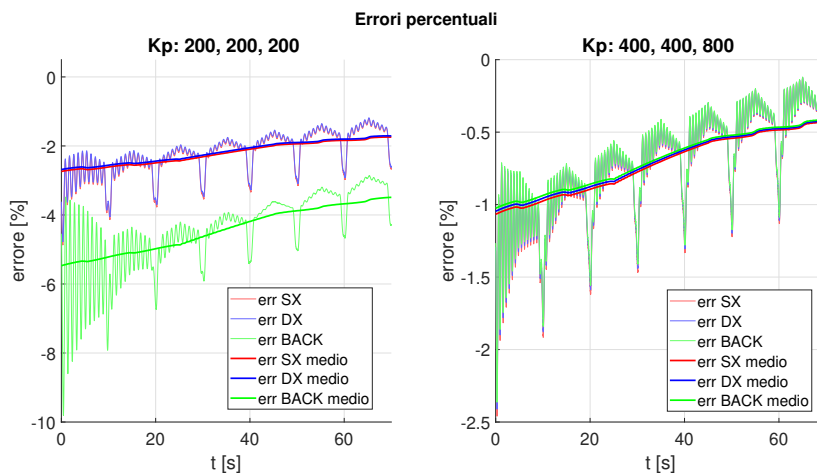


Figura 5.6: Errori di posizione percentuali al variare dei guadagni per tre cavi

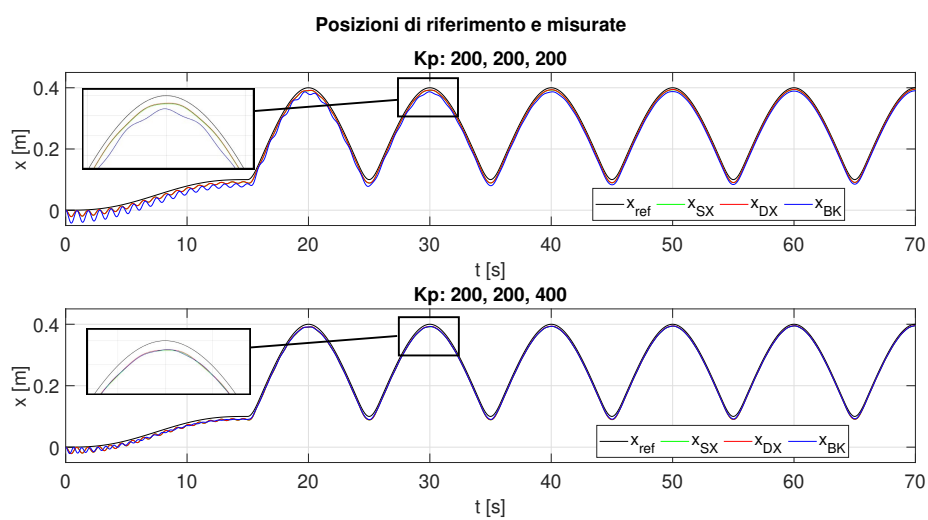


Figura 5.7: Inseguimento di posizione per tre cavi

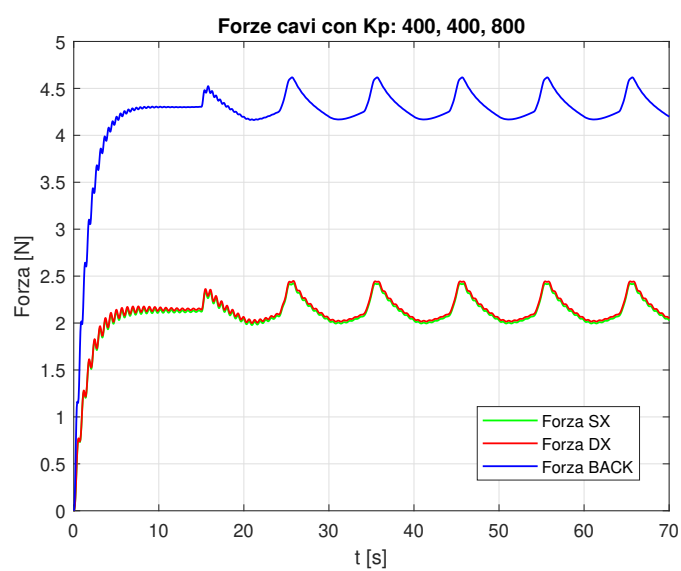


Figura 5.8: Forze risultanti per tre cavi

5.3 Simulazioni giunti

Si cerca ora di studiare dal punto di vista simulativo anche il comportamento dei giunti. Nota la struttura con parallelogramma articolato che caratterizza e distingue MariBot 2.0 dal suo predecessore, si può costruire da una parte il modello che permette l'analisi dinamica diretta del manipolatore, il quale consente di verificare la correttezza delle coppie motrici misurate sperimentalmente rispetto a quelle calcolate dalla teoria, partendo dalle coordinate libere di giunto, e dall'altra le equazioni per la cinematica inversa del robot, che permettono di determinare le posizioni angolari dei link o dei giunti in funzione della posizione dell'end-effector.

5.3.1 Modello del sistema

Il modello del MariBot 2.0 dal punto di vista dei soli giunti, può essere ottenuto attraverso approccio energetico di Lagrange.

Dal momento che il manipolatore è piano l'energia potenziale è nulla; pertanto, la Lagrangiana coincide con l'energia cinetica.

$$L = E_k - E_p = E_k \quad (5.6)$$

Si considerano come coordinate libere del sistema quelle attuate, cioè gli angoli θ_1 e θ_3 di LINK1 e LINK3.

In assenza del contributo dei cavi, le coppie motrici richieste devono essere tali da equilibrare le forze esterne agenti sui due membri C_1 e C_3 , calcolate sfruttando la nota equazione energetica:

$$\begin{cases} \frac{\partial}{\partial t} \left(\frac{\partial E_k}{\partial \dot{\theta}_1} \right) - \frac{\partial E_k}{\partial \theta_1} = C_1 \\ \frac{\partial}{\partial t} \left(\frac{\partial E_k}{\partial \dot{\theta}_3} \right) - \frac{\partial E_k}{\partial \theta_3} = C_3 \end{cases} \quad (5.7)$$

Tali coppie corrispondono, dunque, a quelle presenti ai giunti del manipolatore e, quindi, dovranno essere riportate al motore considerando i rapporti di trasmissione e i rendimenti presenti, come sarà descritto oltre.

Si calcola dunque l'energia cinetica attraverso il teorema di Koenig. Avendo cinque membri l'energia cinetica totale sarà la somma dell'energia cinetica dei singoli membri. I simboli impiegati fanno riferimento alla tabella riportata di seguito.

Grandezza	Valore	Descrizione
l_0	0.5 m	Lunghezza LINK0 (telaio)
l_1	0.304 m	Lunghezza LINK1
l_2	0.460 m	Lunghezza LINK2
l_3	0.05 m	Lunghezza LINK3 (azionato da motore 3)
l_4	0.304 m	Lunghezza LINK4 (parallelo a LINK1)
l_5	0.05 m	Lunghezza LINK5 (parallelo a LINK3)
m_{L0}	0.850 kg	Massa LINK0
m_{L1}	2.93 kg	Massa LINK1
m_{L2}	2.35 kg	Massa LINK2
m_{L3}	0.01 kg	Massa LINK3
m_{L4}	0.064 kg	Massa LINK4
m_{L5}	0.01 kg	Massa LINK5
I_{L1}	0.04 kgm ²	Inerzia al baricentro LINK1
I_{L2}	0.0645 kgm ²	Inerzia al baricentro LINK2
I_{L3}	2.08x10 ⁻⁶ kgm ²	Inerzia al baricentro LINK3
I_{L4}	0.04 kgm ²	Inerzia al baricentro LINK4
I_{L5}	2.08x10 ⁻⁶ kgm ²	Inerzia al baricentro LINK5
G_1	0.142 m	Distanza J1-baricentro LINK1
G_2	0.2 m	Distanza J2-baricentro LINK2
G_3	0.025 m	Distanza J1-baricentro LINK3
G_4	0.152 m	Distanza estremo LINK3-baricentro LINK4
G_5	0.025 m	Distanza J2-baricentro LINK5
I_{m1}	5.3x10 ⁻⁵ kgm ²	Inerzia motore 1
I_{m3}	5.3x10 ⁻⁵ kgm ²	Inerzia motore 3
η_{m1}	0.7	Rendimento riduttore 1
η_{m3}	0.7	Rendimento riduttore 3
i_{m1}	8	Rapporto di trasmissione riduttore 1
i_{m3}	8	Rapporto di trasmissione riduttore 3
F_{c1}	0.01	Coeff. attrito coulombiano riduttore 1
F_{c3}	0.003	Coeff. attrito coulombiano riduttore 3
F_{v1}	0.024 Ns/m ²	Coeff. attrito viscoso riduttore 1
F_{v3}	0.003 Ns/m ²	Coeff. attrito viscoso riduttore 3

LINK1: rotazione attorno al giunto uno.

$$E_{k,1} = \frac{1}{2} \left(m_{L,1} v_{G_1}^2 + I_{L1}^{(G_1)} \dot{\theta}_1^2 \right)$$

LINK3: rotazione attorno al giunto uno.

$$E_{k,3} = \frac{1}{2} \left(m_{L,3} v_{G_3}^2 + I_{L3}^{(G_3)} \dot{\theta}_3^2 \right)$$

LINK4: rototraslazione

$$E_{k,4} = \frac{1}{2} \left(m_{L4} v_{G_4}^2 + I_{L4}^{(G_4)} \dot{\theta}_1^2 \right)$$

LINK5: rototraslazione

$$E_{k,5} = \frac{1}{2} \left(m_{L,5} v_{G_s}^2 + I_{LS}^{(G_s)} \dot{\theta}_3^2 \right)$$

LINK2: rototraslazione

$$E_{k,2} = \frac{1}{2} \left(m_{L,2} v_{G_2}^2 + I_{L2}^{(G_2)} \dot{\theta}_3^2 \right)$$

L'energia cinetica totale risulta quindi:

$$E_k = E_{k,1} + E_{k,2} + E_{k,3} + E_{k,4} + E_{k,5}$$

Sostituendo all'interno della 5.7 è possibile dunque ricavare le coppie ai giunti 1 e 3 richieste.

Da queste si riporta alle coppie motrici secondo la:

$$\begin{cases} C_{M_1} = C_1 \tau_1 + I_{M_1} \ddot{\theta}_{M_1} + F_{v,M_1} \dot{\theta}_{M_1} + \frac{I_{P_1} \ddot{\theta}_{M_1} \tau_1^2}{\eta_{M_1} \eta_{P_1}} \\ C_{M_3} = C_3 \tau_2 + I_{M_2} \ddot{\theta}_{M_2} + F_{v,M_2} \dot{\theta}_{M_2} + \frac{I_{P_1} \ddot{\theta}_{M_2} \tau_2^2}{\eta_{M_2} \eta_{P_2}} \end{cases} \quad (5.8)$$

Per completare lo studio delle equazioni impiegate nei modelli simulativi dei giunti, si riportano le formulazioni impiegate per l'analisi cinematica inversa di MariBot 2.0.

Nelle implementazioni utilizzate, infatti, per simulare l'acquisizione dei punti da parte del terapeuta si fornisce graficamente la rappresentazione dello spazio di lavoro dei link in funzione della configurazione di braccio scelta. A questo punto

sarà possibile selezionare con il puntatore gli n_{pti} di passaggio desiderati. Ogni punto è descritto dalle coordinate cartesiane nel piano XY avente origine nel giunto 1 e che rappresentano la posizione terminale del LINK2 $P(x, y)$, nella mezzeria della travetta di struttura per i passacavi anteriori.

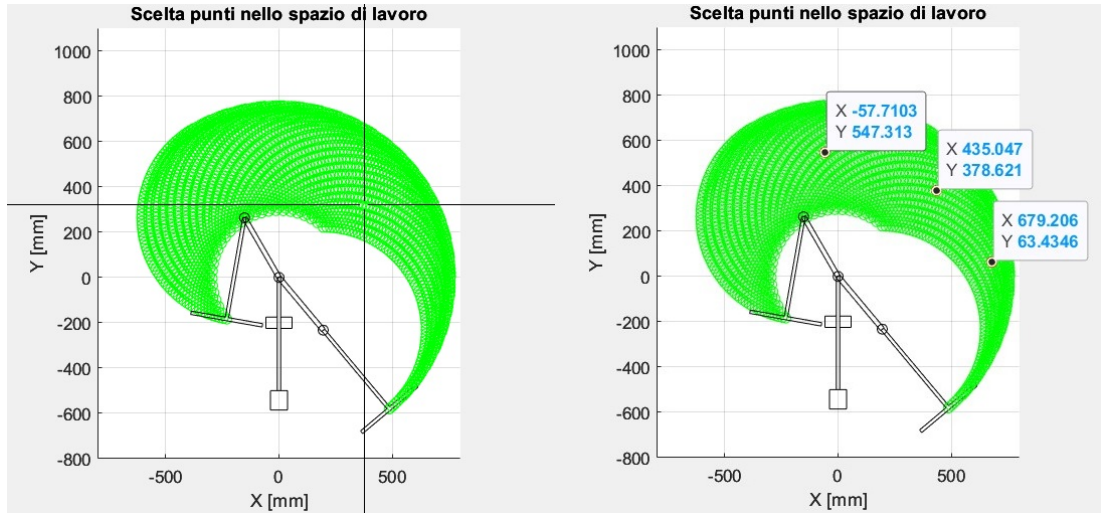


Figura 5.9: Selezione dei punti di passaggio per la simulazione nel caso di configurazione a braccio destro

Per risalire dunque alle coordinate libere θ_1 e θ_3 occorre sfruttare le note formule per la cinematica inversa di un manipolatore scara a due membri che, nel nostro caso, permettono di ricavare le coordinate di rotazione del giunto 1 e del giunto 2 (θ_1 e θ_2)

Sarà poi possibile ricondursi alle coordinate libere sfruttando le relazioni tra i diversi angoli.

Seguendo la nomenclatura per gli angoli riportati in Figura 5.10 e definendo come "lefty" un valore pari a 0 per la configurazione a braccio destro o 1 se invece in braccio sinistro, si ricava la procedura di cinematica inversa, composta dei passaggi che seguono:

$$\theta_2 = (-1)^{lefty} \cdot \arccos(x^2 + y^2 - l_1^2 - l_2^2) / (2l_1l_2) \quad (5.9)$$

$$\beta = (\arccos(l_1 + l_2^2 \cos(\theta_2) / \sqrt{x^2 + y^2})) \quad (5.10)$$

$$\theta_1 = \text{atan2}(y, x) - ((-1)^{lefty})\beta \quad (5.11)$$

$$\theta_3 = \theta_1 + \theta_2 + \alpha \quad (5.12)$$

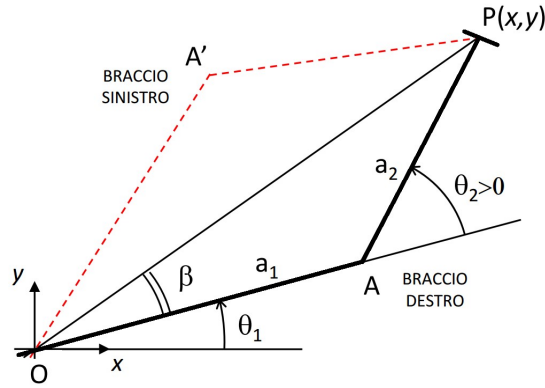


Figura 5.10: Angoli impiegati per la cinematica inversa

5.3.2 Risultati simulativi giunti

Attraverso le relazioni di cinematica inversa e di dinamica diretta, tramite script *MATLAB* si ricavano le coppie teoriche richieste ai motori posizionate ai giunti per l'esecuzione del movimento desiderato secondo punti di passaggio acquisiti. La pianificazione di traiettoria segue le medesime leggi utilizzate per i cavi.

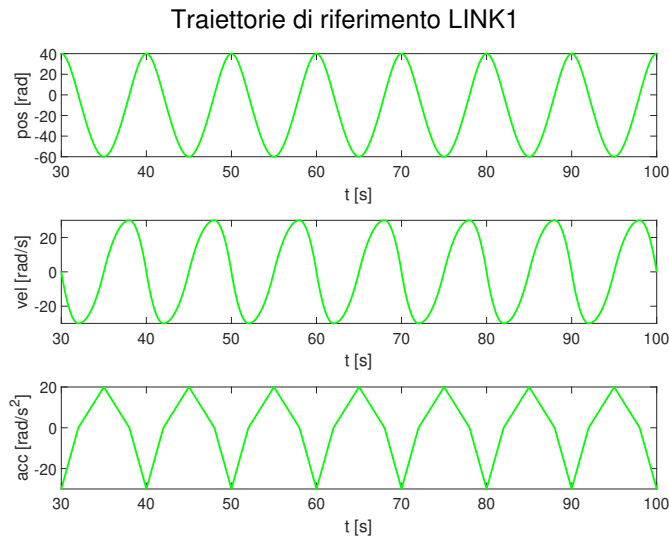


Figura 5.11: Riferimenti per LINK1

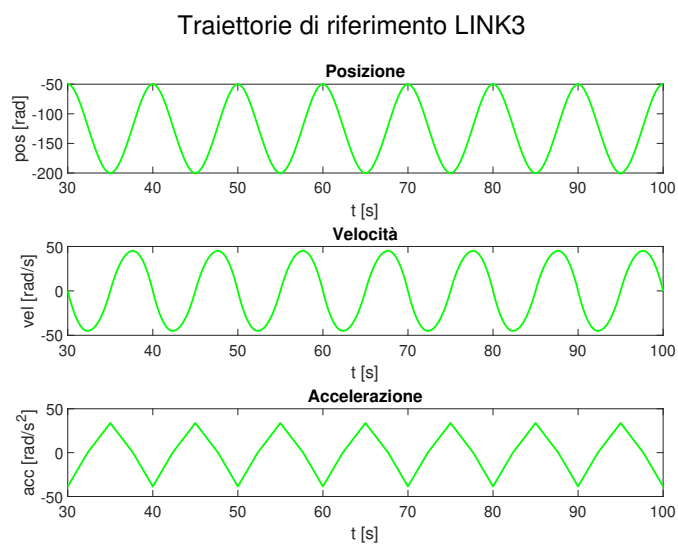


Figura 5.12: Riferimenti per LINK3

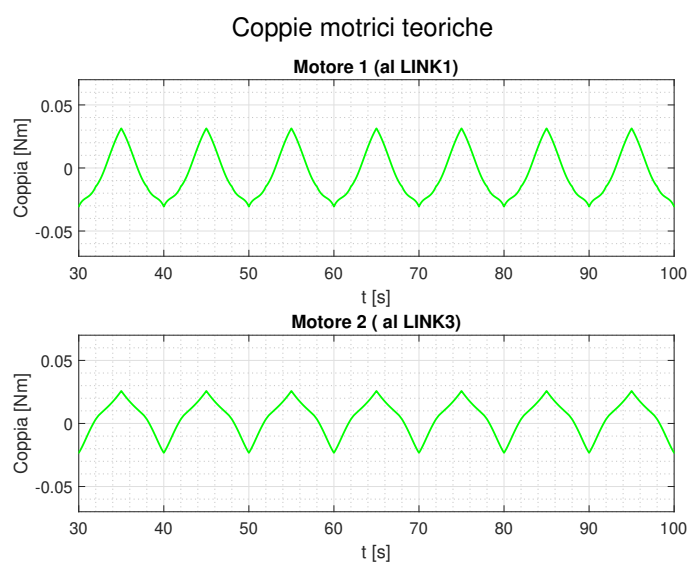


Figura 5.13: Coppie motrici richieste

5.4 Risultati sperimentali

Le prove sperimentali per il controllo a basso livello sono state eseguite sul manipolatore sfruttando le medesime modalità impiegate nelle simulazioni.

Ciò che si vuole osservare, quindi, è la corrispondenza tra le coppie e forze teoriche e quelle effettivamente misurate oltre che la variazione del comportamento quando si modificano i guadagni dei controllori.

5.4.1 Risultati sperimentali cavi

Si riportano in questa sezione i risultati ottenuti per le prove sperimentali di movimentazione ciclica del singolo cavo con massa e a tre cavi con ortesi.

- **Singolo cavo**

I guadagni del controllore PID impiegati sull'apparato reale sono gli stessi utilizzati per le prove simulative pertanto, come ci si può aspettare, si osserva un errore nell'inseguimento che si riduce all'aumentare del K_p e K_i , con un ordine di grandezza percentuale del tutto confrontabile con quanto osservato nella sezione precedente.

Anche la forza necessaria per il movimento si assesta effettivamente nell'intorno dei 7N, ad indicare che la modellizzazione del sistema reale utilizzata nelle simulazioni è effettivamente sensata. Si osserva, tuttavia, come le oscillazioni reali abbiano un'ampiezza maggiore a causa delle ulteriori non idealità presenti nel sistema reale oltre a quelle rappresentate dal rendimento del motore.

- **Tre cavi** Le osservazioni fatte per le prove sul singolo cavo sono valide anche per il movimento dell'ortesi con tre cavi.

Ciò che si nota è la effettiva differenza di forza richiesta tra i cavi anteriori e quello posteriore supposta in fase di simulazione: ad esso è infatti richiesto uno sforzo maggiore per eseguire la traiettoria desiderata.

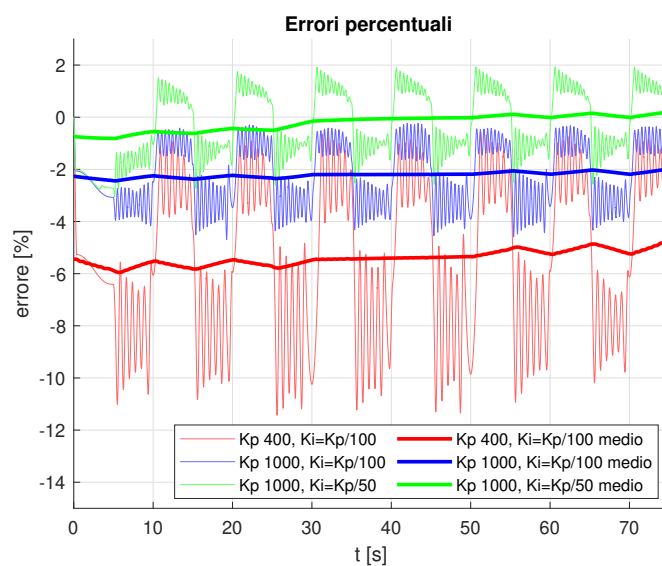


Figura 5.14: Errori di posizione percentuali al variare dei guadagni per singolo cavo

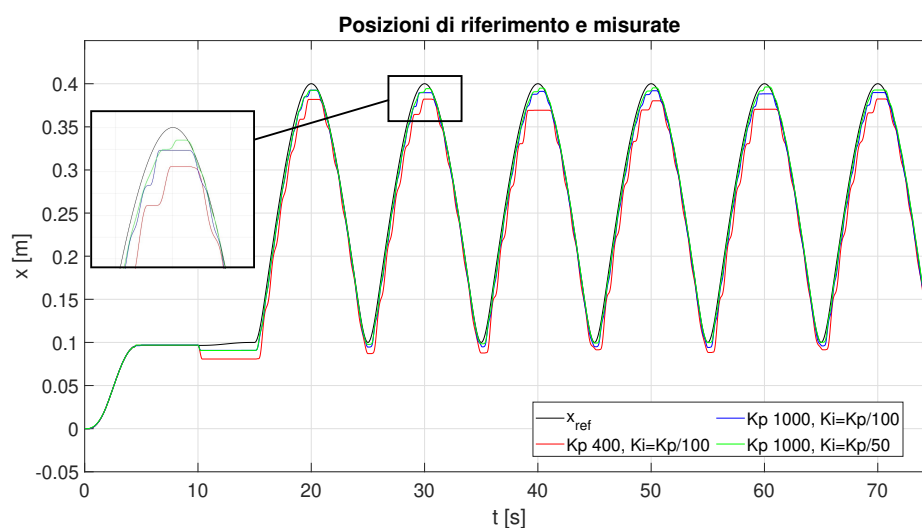


Figura 5.15: Inseguimento di posizione per singolo cavo

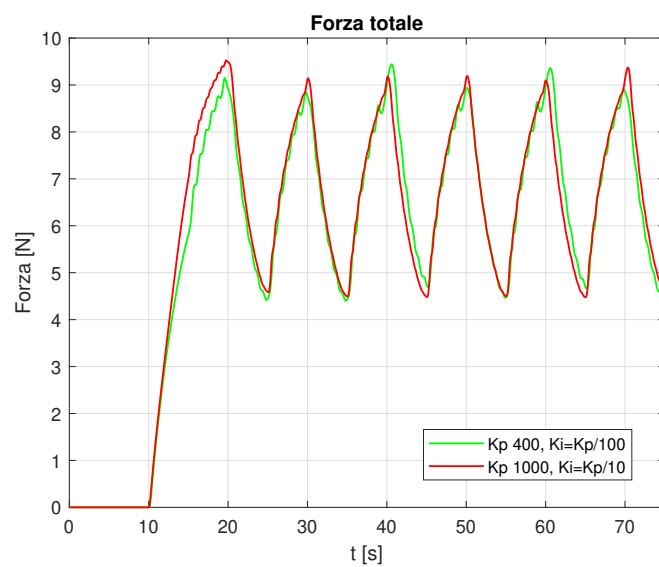


Figura 5.16: Forza risultante per singolo cavo

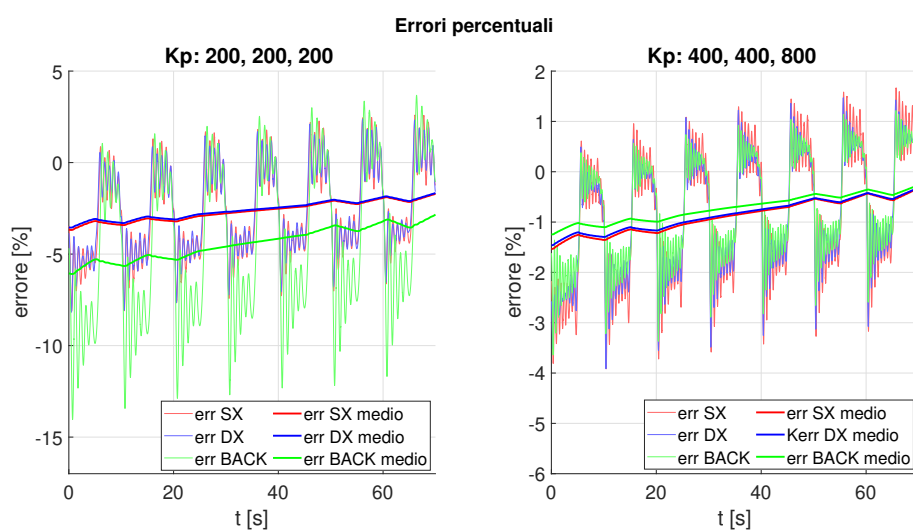


Figura 5.17: Errori di posizione percentuali al variare dei guadagni per tre cavi

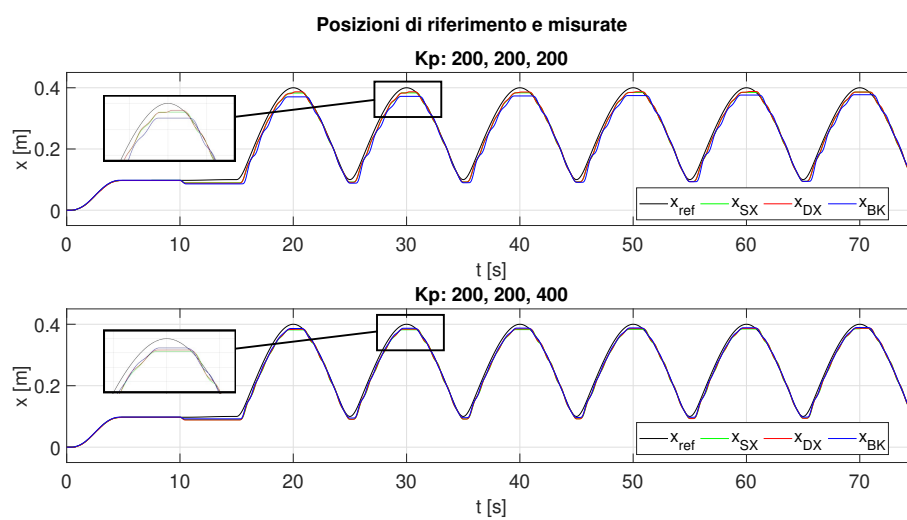


Figura 5.18: Inseguimento di posizione per tre cavi

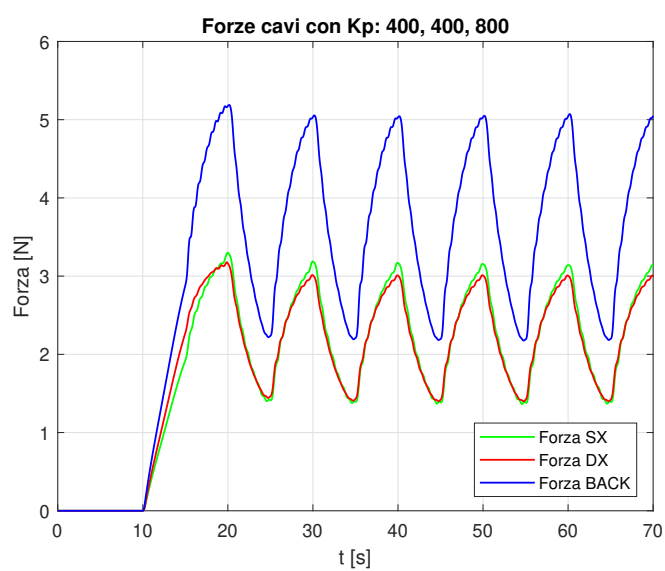


Figura 5.19: Forze risultanti per tre cavi

5.4.2 Risultati sperimentali giunti

Operando in modo analogo a quanto visto per i cavi, si riportano i risultati sul controllo a basso livello per le traiettorie dei giunti. In questo caso si fa riferimento al solo caso di inseguimento "ottimale", cioè effettuato con una taratura spinta dei guadagni del controllore PID in modo da ridurre quanto più possibile il valore dell'errore di inseguimento.

Tale obiettivo è stato ottenuto per $Kp_1=3.5$, $Ki_1=Kp_1/50$, $Kd_1=Kp_1/200$ per quanto riguarda il controllo del LINK1 e $Kp_3=5$, $Ki_3=Kp_3/200$, $Kd_1=Kp_1/200$. Si può osservare come sia richiesto un maggior sforzo al controllore del giunto 3: il motivo di tale necessità risiede nel fatto che le masse dei link sono elevate e il loro contributo durante una movimentazione ciclica, soprattutto se ad alte velocità, è notevole. Soprattutto per quanto riguarda il LINK2 (azionato dal motore 3), il cui movimento risentirà anche della rotazione del LINK1.

Si riportano di seguito i risultati ottenuti.

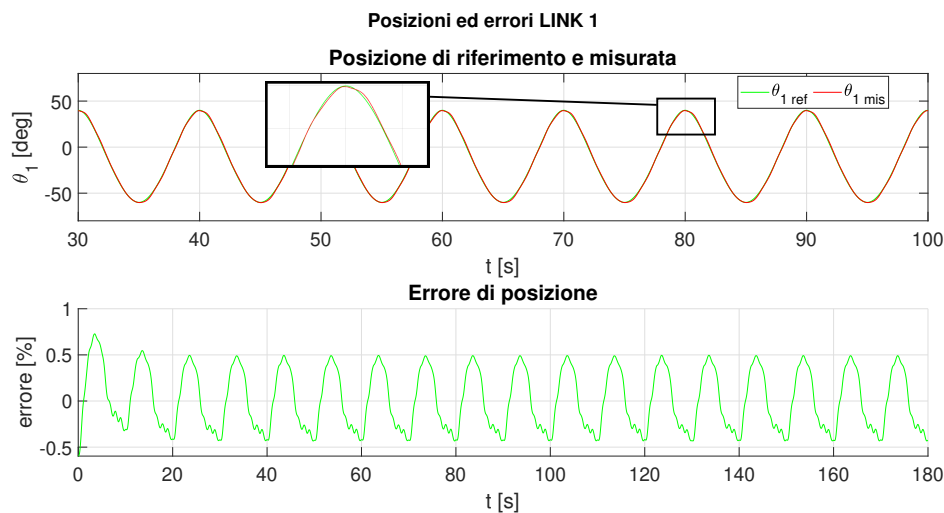


Figura 5.20: Inseguimento di posizione LINK1

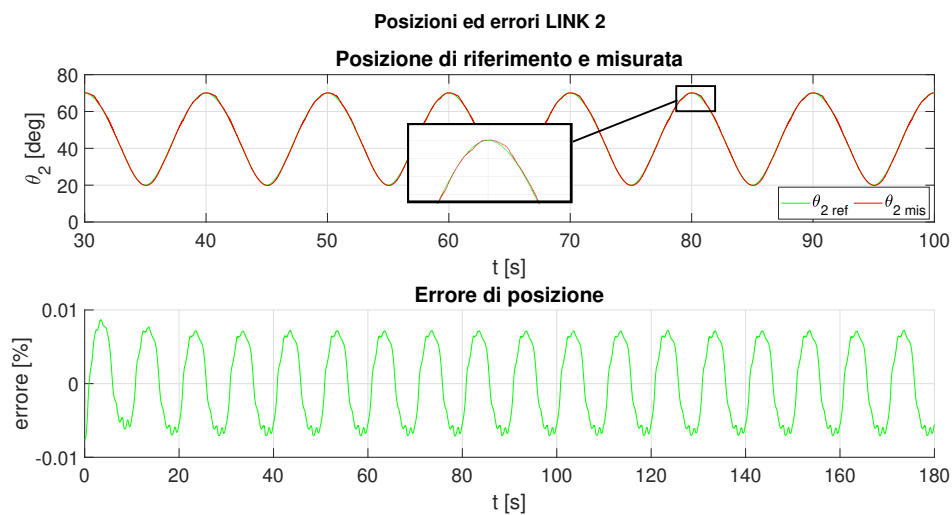


Figura 5.21: Inseguimento di posizione LINK2

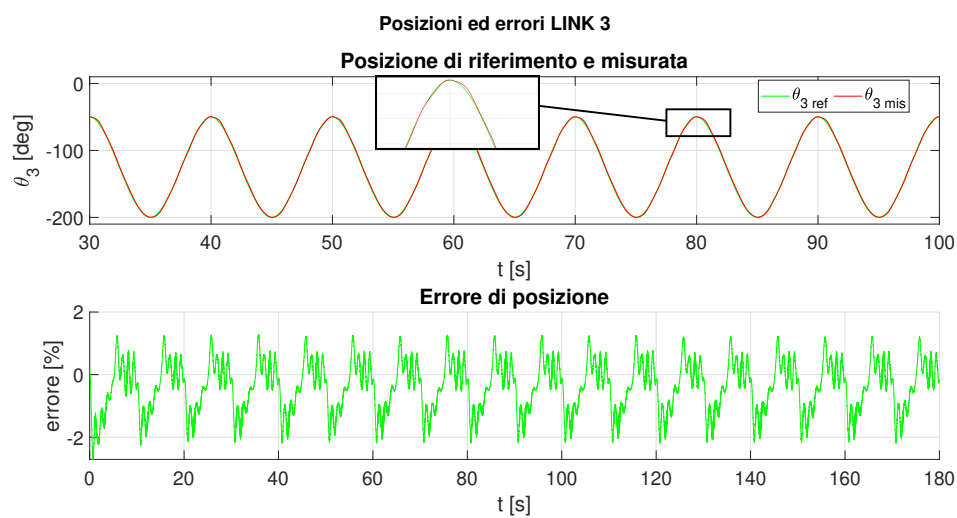


Figura 5.22: Inseguimento di posizione LINK3

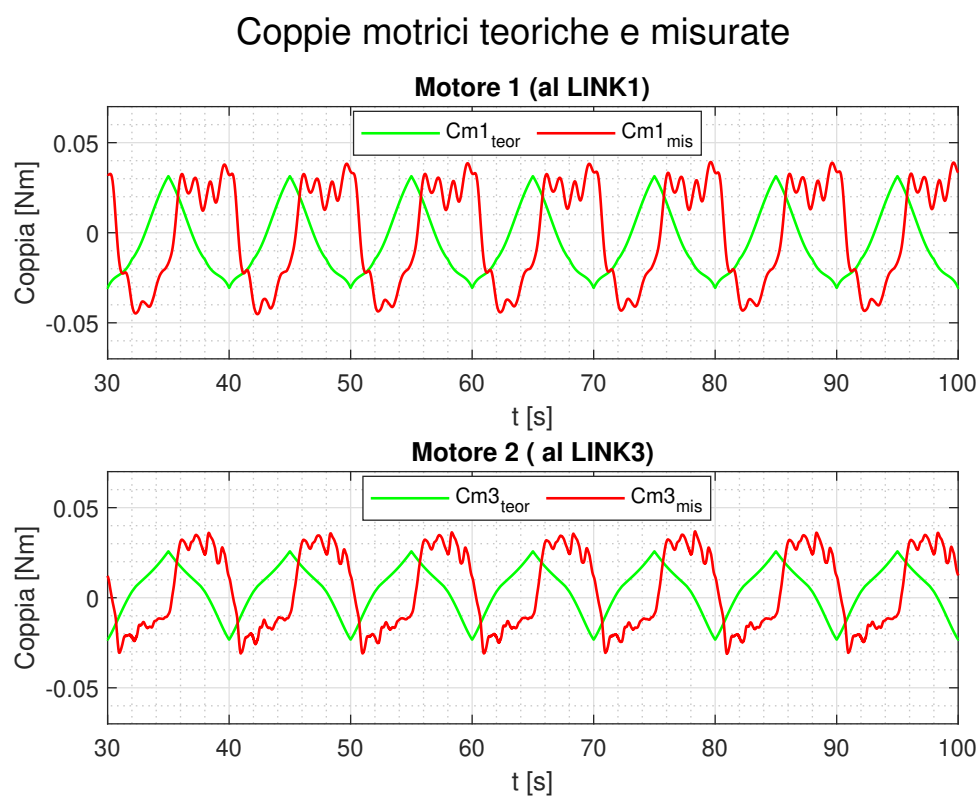


Figura 5.23: Confronto tra coppie motrici e sperimentali

Capitolo 6

Controllo ad alto livello

L'utilizzo di controllori di questo tipo, cioè capaci di modificare la propria risposta in base all'azione del paziente, fa sì che l'interazione uomo-macchina risulti notevolmente maggiore rispetto a robot comandati da semplici PID e, pertanto, rende possibile ottenere assistenza diversa al variare del tempo e del soggetto cui è destinata la riabilitazione.

In questo capitolo verrà esposta la teoria alla base del controllo adattativo, verrà fornita la descrizione dell'implementazione usata nel modello del MariBot2.0 basata sul concetto di forgetting factor, già proposto in [22] e infine saranno riportati i risultati ottenuti a valle delle simulazioni.

6.1 Controllo adattativo con forgetting factor

La logica che gestisce ognuno dei motori impiegati nel manipolatore vede la cooperazione di un controllore PD (proporzionale-derivativo) in parallelo ad un controllore adattativo con forgetting factor.

La collaborazione di questi due contributi determina la forza complessiva che il robot fornisce per inseguire il riferimento e che può quindi essere espressa come segue:

$$F_{robot} = F_{static} + F_{adapt} = -K_p \cdot e - K_d \cdot \dot{e} + \mathbf{Y} \cdot \mathbf{a} \quad (6.1)$$

dove $e = \theta - \theta_{ref}$ è l'errore di posizione mentre $\dot{e} = \dot{\theta} - \dot{\theta}_{ref}$ quello di velocità.

Il contributo dell'adattativo è dato dal prodotto tra $\mathbf{Y}$, vettore dei regressori e $\mathbf{a}$, vettore dei pesi stimati.

Queste grandezze rappresentano il cuore del controllore adattativo e sono direttamente connesse al concetto delle *Radial Basis Functions*, o RBF.

6.1.1 Radial Basis Functions

Una generica RBF (funzione radiale di base gaussiana) $g_j(x)$ è una funzione matematica descritta dalla 6.2. Il suo valore dipende dalla distanza (tipicamente calcolata come norma Euclidea) tra l'ingresso x e un punto appartenente al dominio denominato centroide μ_j ; la sua ampiezza è definita dal parametro σ detto deviazione standard o indicatore di *smoothing*.

$$g_j(x) = \exp\left(\frac{-||x - \mu_j||^2}{2\sigma^2}\right) \quad (6.2)$$

Nel caso applicativo di MariBot 2.0, i centroidi vengono posizionati in modo che vi sia una suddivisione equispaziata in p punti del campo di lavoro del singolo motore mentre l'argomento x della funzione corrisponde alla posizione misurata dagli encoder.

Per quanto riguarda la deviazione standard, come suggerito da [22], essa viene definita come $\sigma = \alpha \cdot \Delta x$, dove $\alpha \in [0.75, 1]$ e Δx è la distanza tra un centroide e l'altro.

Il valore assunto da ciascuna j -esima delle p RBF che discretizzano il range di movimento, calcolato in corrispondenza di una determinata misura di posizione x , va a costituire uno degli elementi del vettore dei regressori Y ($1 \times p$). Evidentemente, quindi, la matrice Y verrà aggiornata ad ogni nuova misurazione di x .

$$\mathbf{Y} = [g_1(j), \dots, g_p(j)]$$

Il vettore a ($p \times 1$) contiene dei coefficienti di pesatura calcolati secondo la 6.3 che, analiticamente, definiscono la forma delle diverse gaussiane ma che possono essere interpretati come degli indicatori delle capacità di movimento del paziente.

Il prodotto Ya , cioè il contributo in forza fornito dal controllo adattativo, è ottenuto in modo che gli elementi di a vadano ad alterare le ampiezze di Y , dunque di ciascuna gaussiana. Applicativamente questo significa che, dato un paziente con capacità definite dalla stima di a , considerando la posizione dello spazio di lavoro nella quale ci si trova in un determinato istante x , in relazione con quella dei centroidi imposti, sarà possibile pesare diversamente ciascuna delle p RBF in modo da alterare il valore del prodotto Ya con conseguente adattamento della forza erogata dal manipolatore.

Si riporta in Figura 6.1 un grafico d'esempio che permette di comprendere appieno il comportamento appena illustrato.

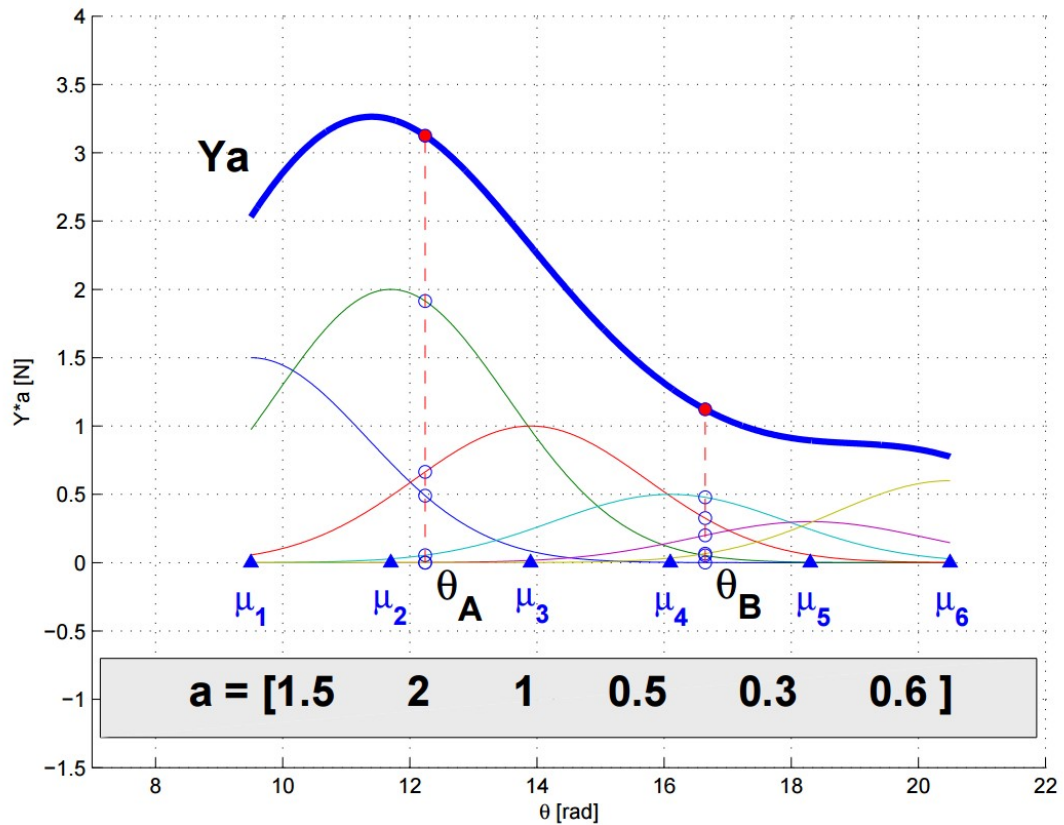


Figura 6.1: Esempio di prodotto Ya e gaussiane pesate dal vettore a

Il grafico, che rappresenta un'istantanea di un determinato intervallo temporale durante la terapia, riporta sull'asse delle ascisse la posizione misurata x (considerata, a titolo d'esempio, come la posizione angolare dei motori dei link

θ [rad]) e sull'asse delle ordinate la il prodotto Ya , ossia il contributo di forza fornito dal controllo adattativo F_{adapt} [N].

Nel medesimo grafico è inoltre raffigurato l'andamento delle gaussiane localizzate nei centroidi di discretizzazione per quello specifico range degli ingressi, in questo caso nominati $\mu_1, \mu_2, \dots, \mu_6$. Il vettore delle stime a assume i valori d'esempio riportati in figura 6.1.

Si può quindi intuire come i diversi valori assunti da tale vettore vadano ad alterare la forma della RBF ad essi associata: in questo senso l'elemento a_2 , maggiore, farà assumere valori più elevati alla corrispondente gaussiana centrata in μ_2 , mentre l'elemento minore, a_5 , porterà a valutare con un peso minore la corrispondente RBF in μ_5 .

Una volta stabilita l'ampiezza delle diverse gaussiane in base ad a , si sfrutta questo comportamento matematico per determinare il contributo in forza Ya che il controllore adattativo deve fornire in funzione della posizione nella quale ci si trova in un preciso istante.

Nel caso in cui ci si trovi in θ_A , per esempio, le gaussiane con un peso maggiore saranno quelle con centroidi più vicini a θ_A stesso, cioè μ_2 e μ_3 ; trovandosi invece in θ_B è evidente invece che le stesse RBF avranno un incidenza minore nel computo del prodotto Ya , che sarà comunque dato dalla somma dei contributi delle p gaussiane per quella posizione.

Si può anche osservare come il parametro σ sia molto influente in questa fase. Un σ elevato porta a gaussiane più ampie, il cui contributo viene pertanto smorzato in maniera più graduale e che può essere risentito anche se lontane dal punto: per ottenere una variazione più evidente del contributo di forza risultante occorre compensare con pesature di a molto diverse tra le varie gaussiane. Al contrario con deviazioni standard basse si otterranno RBF più strette, portando il prodotto Ya a dipendere fortemente dalla gaussiana più vicina.

6.1.2 Vettore dei pesi stimati

Come accennato in precedenza, il vettore a definisce l'entità dell'infermità del paziente in un determinato intervallo dello spazio di lavoro: se i pesi stimati di a

sono bassi significa che il paziente è, in quel tratto, capace di eseguire in maniera autonoma il movimento. Al contrario, nei tratti di esercizio in cui esso risulta più debole, i pesi assumeranno un valore più elevato, risultando in un maggior contributo richiesto ad F_{adapt} .

Questa è la potenza del controllore adattativo implementato in MariBot 2.0 che è quindi in grado di adattare la sua risposta analizzando di continuo la risposta del paziente e riconoscendo in quali tratti dell'esercizio occorre fornire una maggior assistenza.

La stima dei pesi, aggiornata in tempo reale, viene calcolata come somma algebrica di due contributi secondo la legge:

$$\dot{a} = \dot{a}_s + \dot{a}_f = -\Gamma^{-1}Y^T s - \frac{1}{\tau}Y^T(YY^T)^{-1}Ya \quad (6.3)$$

Di seguito una descrizione di ciascun termine.

- Sliding control $\dot{a}_s$

Il termine

$$\dot{a}_s = -\Gamma^{-1}Y^T s \quad (6.4)$$

adatta i pesi in modo da minimizzare la *sliding surface* s (6.5), definita come combinazione lineare degli errori di posizione e velocità, valutati diversamente in funzione del parametro Λ :

$$s = \dot{e} + \Lambda e \quad (6.5)$$

Attraverso la 6.4 il robot aggiorna i parametri di a in modo da poter esercitare la corretta forza per far rimanere il paziente nella traiettoria desiderata. Ciò significa che, con una rapidità nell'aggiornamento imposta dal parametro Γ , l'errore in posizione e tenderà ad annullarsi così come il valore di s . Il raggiungimento di questa condizione sta ad indicare che il controllore ha effettivamente adattato la sua risposta in funzione della dinamica sviluppata dal paziente.

- Assist-as-needed control $\dot{a}_f$

Il secondo termine è descritto dalla 6.3 ed è detto di *forgetting* o *assist-as-needed*.

$$\dot{a}_f = -\frac{1}{\tau} Y^T (Y Y^T)^{-1} Y a \quad (6.6)$$

L'importanza di tale contributo risalta nel momento in cui si verifica il tracking perfetto, cioè quando $\dot{a}_s$ ha agito per annullare la sliding function, garantendo l'inseguimento della traiettoria in maniera esatta. Occorre evidenziare che tale obiettivo viene comunque raggiunto, indipendentemente dall'entità del contributo fornito dal soggetto sotto terapia.. Una volta raggiunta questa condizione è desiderabile che il manipolatore diminuisca la sua assistenza, in modo tale da richiedere un maggior sforzo al paziente così da rendere la terapia più efficace.

Seguendo le orme di Reinkensmeyer e Bobrow viene quindi inserito il termine $\dot{a}_f$ che definisce la riduzione esponenziale della forza esercitata dal controllo adattativo secondo una costante τ : essa è la costante di tempo del processo di forgetting, con il quale il controllore dimentica la dinamica precedentemente appresa. Per rendere ancor più evidente questo comportamento si definisce il *forgetting factor* ff come l'inverso di τ : un ff grande indica un manipolatore che dimentica molto rapidamente la dinamica appresa; al contrario un valore piccolo di questo paramtro definisce un controllore più severo.

Il termine di decadenza non è dato solo da τ ma dipende fortemente dal valore assunto dal vettore dei regressori Y : la riduzione dell'output robot sarà tanto maggiore quanto più vicini ci si trova alle RBF di discretizzazione. Ciò significa che il controllo implementato, di base, tende a "premiare" il paziente che riesce a portarsi sufficientemente vicino al riferimento e a "esortarlo a fare meglio" riducendo il proprio contributo assistenziale.

Occorre evidenziare, tuttavia, che diminuendo la componente di forza data dallo sliding control, non si sarà più in grado di annullare interamente l'errore di posizione; ciò fa sì che il controllore implementato conceda al paziente una maggior possibilità di interazione ma tollerando un certo errore sull'inseguimento.

Per concludere si osserva che i contributi descritti finora hanno interessi contrastanti: il termine $\dot{a}_s$ determina lo sforzo apportato dal robot per garantire il

tracking perfetto, mentre $\dot{a}_f$ mira a una partecipazione attiva del paziente, tollerando un errore di posizione diverso da zero. Un corretto settaggio dei parametri del controllore adattativo deve dunque tenere in considerazione le capacità motorie e di apprendimento del paziente col fine di garantire un compromesso tra precisione nell'esecuzione del movimento e partecipazione del soggetto. Si può, per esempio, preferire un controllore "accondiscendente" che quindi premia gli sforzi del paziente tollerando errori di posizione anche elevati oppure più "severo" che in tutti i casi cerca di imporre il movimento di riferimento.

6.1.3 Schema di controllo

L'implementazione della logica di controllo adattativo appena descritta si può ottenere tramite uno schema che risulta valido per diverse applicazioni e che è riportato in Figura 6.2.

Si può osservare come gli ingressi siano rappresentati dagli errori di posizione e velocità calcolati tra i riferimenti ottenuti dal pianificatore di traiettoria e le misurazioni. Tali errori vengono sfruttati nel parallelo composto tra l'architettura statica a basso livello PD e l'alto livello rappresentato dall'adattativo, che come prima operazione richiede il calcolo delle sliding surfaces e del vettore dei regressori Y . Questi permettono dunque di calcolare i due termini che portano alla stima dei vettori pesati a secondo le formulazioni di *sliding control* ($\dot{a}_s$) e *forgetting* ($\dot{a}_f$).

Si ottiene dunque il contributo in forza fornito dall'adattativo $Y a$ che va ad agire sul manipolatore, secondo le relazioni con il paziente determinate dal mondo fisico.

6.2 Simulazione per controllo adattativo del singolo cavo

Nella seguente sezione si riporta a titolo di esempio la descrizione dello schema *Simulink* adottato per la simulazione del controllore adattativo utilizzato per la

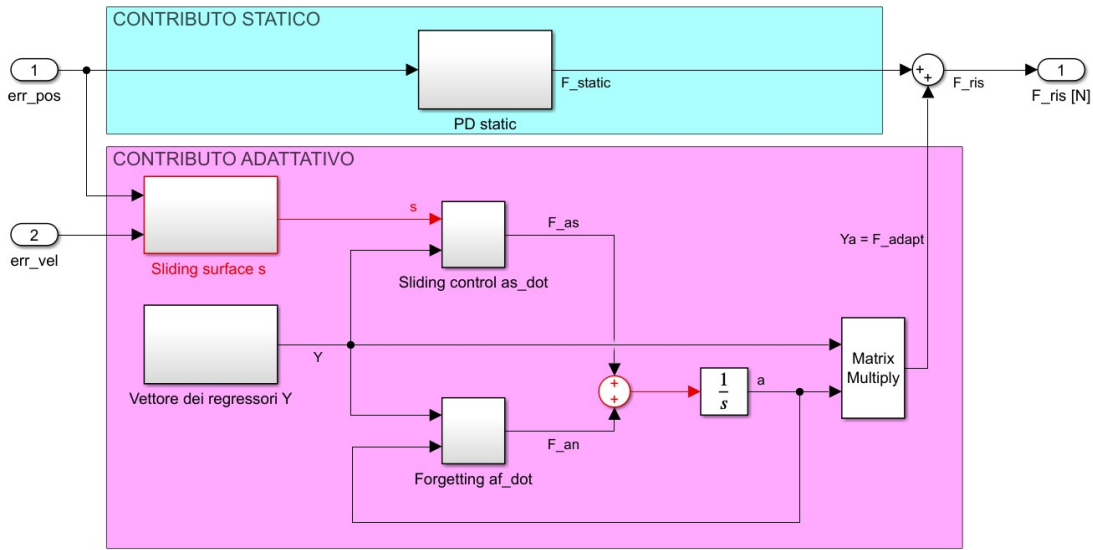


Figura 6.2: Generico schema di controllo in architettura adattativa con forgetting factor

gestione del singolo cavo di MariBot 2.0. L'intento è quello di osservare l'influenza che la variazione dei diversi parametri porta nella risposta del sistema prima di poter andare a utilizzare la stessa logica sul robot.

6.2.1 Schema simulativo di controllo

Lo schema di controllo complessivo implementato per la simulazione può essere visto come costituito da due blocchi principali: il blocco robot e il blocco paziente. Il primo implementa lo schema teorico già visto nella sezione precedente mentre il secondo serve per modellizzare il comportamento di un soggetto in terapia.

La modellizzazione del paziente è stata realizzata come un semplice controllore PD, in linea con la componente statica del controllore adattativo, cioè privata della classica parte integrale: è ben noto che, dal punto di vista del robot, l'azione di correzione dell'errore a regime viene svolta dal termine di *sliding control*. Per evitare quindi che si crei un contrasto tra le logiche per la gestione del robot e di simulazione del paziente, si è optato appunto per la rimozione della parte integrativa anche da quest'ultimo.

Per cercare di migliorare la modellizzazione si è cercato di simulare il comportamento

di un soggetto capace di apprendere l'esercizio da eseguire nel corso del tempo. Per raggiungere tale obiettivo si impone al controllore un valore di Kp_{pz} crescente secondo una certa costante di tempo τ_{pz} .

Ciò che si vuole ottenere è quindi un paziente con capacità finali di movimento definite dal guadagno proporzionale che possono portarlo, insieme all'assistenza del robot, ad inseguire più o meno bene il riferimento. Tuttavia esso non sarà in grado di fornire fin da subito tutta la forza che quel determinato valore di Kp_{pz} porta ma lo farà solo dopo il tempo τ_{pz} .

La legge di crescita del guadagno proporzionale è definita come:

$$Kp_{pz} = Kp0_{pz} \cdot (1 - e^{-\frac{t}{\tau_{pz}}}) \quad (6.7)$$

Dove $Kp0_{pz}$ rappresenta il valore a regime del guadagno proporzionale. Per cercare di riportarsi quanto più possibile alla realtà, si è imposta la relazione tra la costante di tempo di apprendimento e il numero di cicli di esercizio eseguiti: in questo modo si collega la capacità di imparare il movimento terapeutico del paziente al numero di cicli eseguito, cioè al ciclo n -esimo sarà trascorso il tempo τ e il soggetto starà erogando circa il 63% della forza definita dal valore di $Kp0_{pz}$. La relazione tra il parametro n e τ_{pz} è:

$$\tau_{pz} = n \cdot t_{ciclo} + t_{setup} \quad (6.8)$$

A titolo di esempio, è riportato lo schema completo impiegato per la simulazione del comportamento robot-paziente per il singolo cavo. Oltre agli aspetti descritti in precedenza, si vuole ora evidenziare come i segnali "settings" servano semplicemente per decidere quale tipologia di simulazione si vuole eseguire: solo robot PD, solo robot con sola parte di *sliding*, solo robot con controllo adattativo completo, robot e paziente privo di capacità di apprendimento o, al contrario, in grado di farlo. Il blocco di dinamica inversa permette di ottenere l'accelerazione misurata (da cui posizione e velocità, necessarie al controllo) partendo dalle coppie motrici richieste, secondo il modello esposto nel capitolo 5.

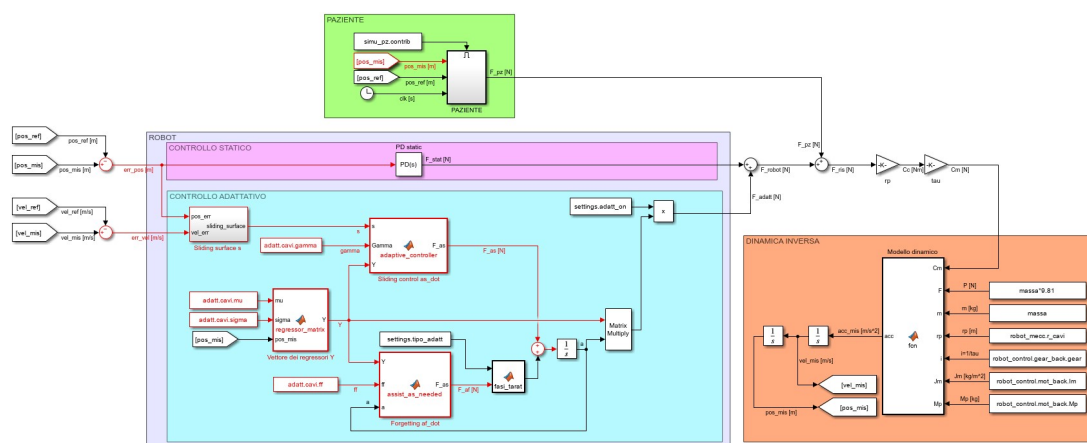


Figura 6.3: Schema di controllo adattativo per le simulazioni nel caso di singolo cavo

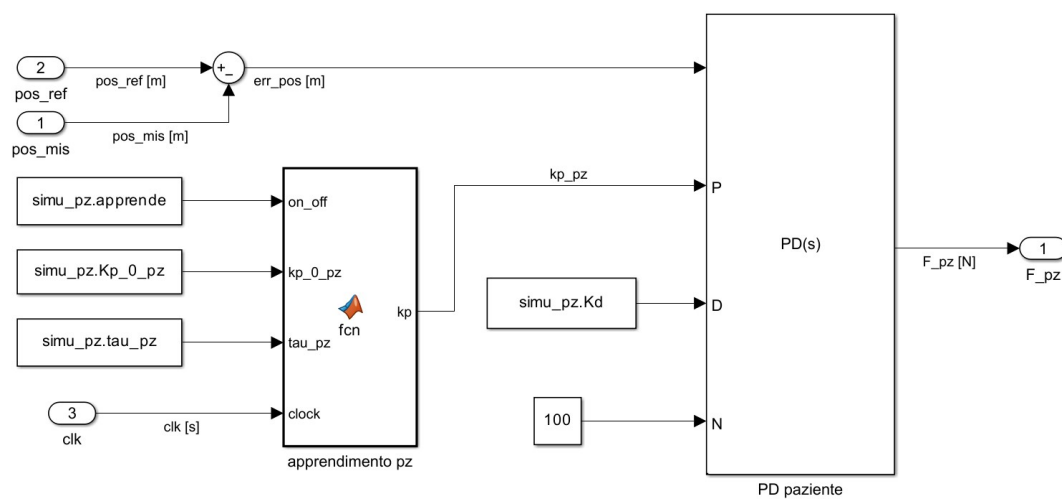


Figura 6.4: Blocco paziente simulato

6.2.2 Risultati simulativi singolo cavo

Per comprendere il significato del contributo dato dalla variazione dei diversi parametri, si riporta ora la procedura in passi seguita per lo svolgimento delle prove simulative nel controllo adattativo del singolo cavo con massa.

La medesima metodologia è stata seguita anche per le prove sperimentali sul MariBot 2.0, sia per quanto riguarda il singolo cavo, che il controllo dei tre cavi con ortesi e infine la gestione dei giunti. I risultati ottenuti saranno riportati nella sezione successiva.

- **Taratura guadagni robot**

Il primo passo consiste nel settaggio dei parametri del controllore del blocco robot. Per prima cosa è stato definito il valore dei guadagni proporzionale e derivativo del PD statico: il criterio impiegato è stato quello di definirli in modo da ottenere un inseguimento sulla posizione con un errore intorno al 20%. Tale condizione è raggiunta quando il manipolatore eroga il minimo di forza in assistenza, perchè data dal solo blocco statico e che, pertanto, definisce il range di libertà di movimento concesso al paziente. In ogni caso, comunque, lo sforzo sarà tale da garantire il mantenimento del peso connesso al cavo con la la forza risultante che si porterà quindi ad oscillare a tale valore (ricordiamo che la massa impiegata per le prove è $m=0.6$ kg).

Si sono quindi imposti: $K_p = 120$ e $K_d=50$.

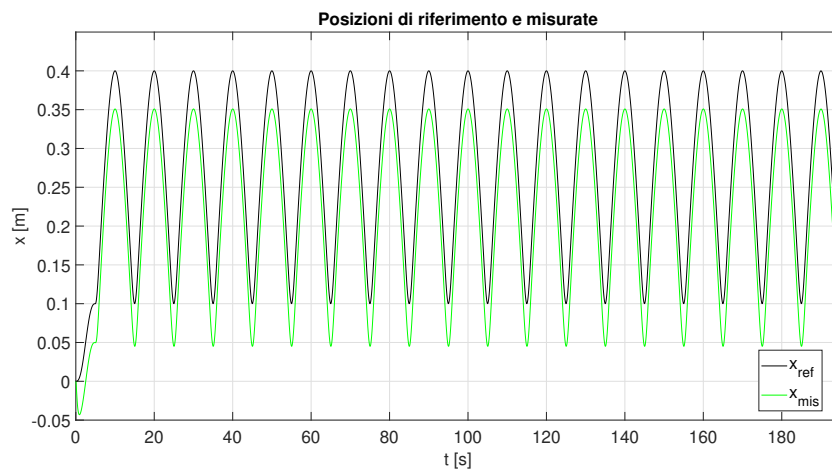


Figura 6.5: Tracking di posizione per controllo statico con $K_p=120$ e $K_d=50$

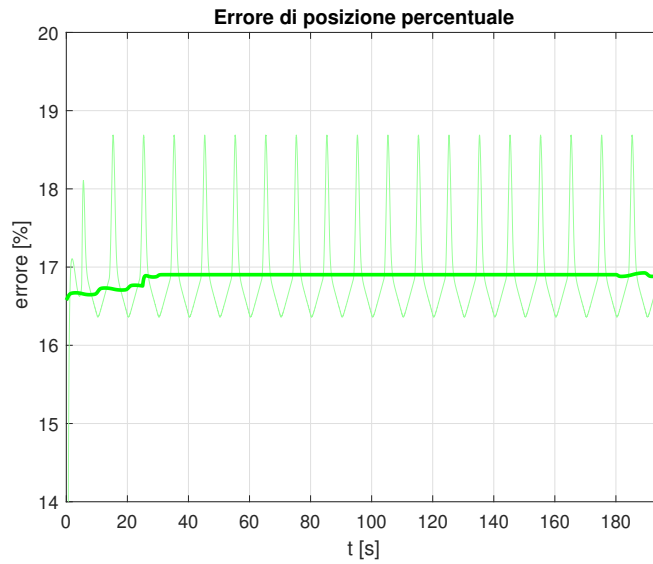


Figura 6.6: Errore di posizione per solo controllo statico con $K_p=120$ e $K_d=50$

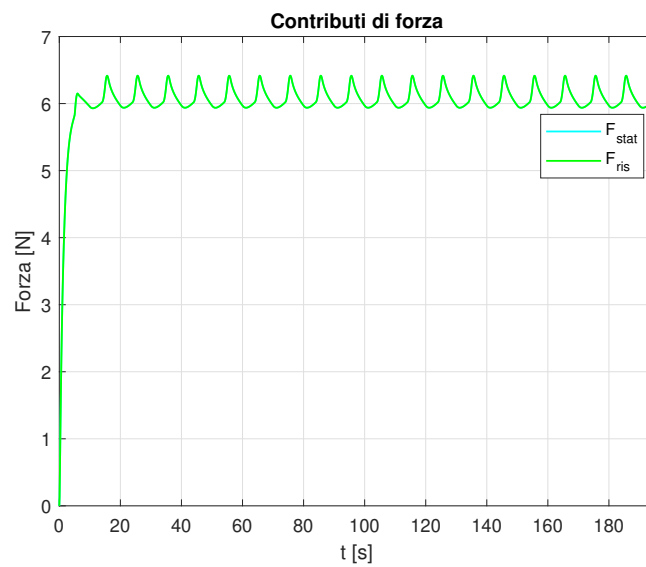


Figura 6.7: Forza risultante per solo controllo statico con $K_p=120$ e $K_d=50$

Si prosegue con la taratura dei parametri dell'adattativo.

Questa è stata distinta nel settaggio iniziale dei parametri della parte di *sliding control*, seguita dunque dalla componente di *forgetting*.

Si mantengono quindi fissi i valori di $\Lambda = 1.5$ e $\sigma = 0.108$. Quest'ultimo è stato definito secondo la logica descritta nella sezione precedente, dunque discretizzando lo spazio di lavoro del singolo cavo, corrispondente alla massima corsa del cavo di 0.43 m (che comprende un margine di tolleranza di 5 cm per evitare che si possa arrivare troppo vicini ai passacavi), in $p=40$ RBF.

A questo punto sono stati provati diversi valori di Γ (Figure 6.8 e 6.9).

In linea con la formulazione di $\dot{a}_s$ vista in 6.4, si osserva come valori più piccoli di tale parametro portino ad ottenere un contributo più rapido dell'azione adattativa, determinato da una maggior velocità nell'aggiornamento dei parametri stimati a cui consegue un più rapido assestamento a zero dell'errore a regime (che parte dal valore residuo concesso dal controllo statico) e della forza nell'intorno del valore di mantenimento noto. Si può quindi osservare che effettivamente l'obiettivo della componente di *sliding* è proprio quello di garantire il tracking perfetto, con una prontezza e intensità inversamente proporzionale al valore di Γ . Il caso di $\Gamma=50$ è un estremo e verifica la condizione per cui l'aggiornamento è così lento da lasciare un errore residuo.

Evidentemente, associata a una crescita della forza adattativa F_{adapt} , vi sarà una diminuzione del contributo del controllore statico F_{static} complementare; per ogni valore dell'*'adaptive gain* Γ la loro somma rappresenta la risultante fornita dal robot per garantire il movimento desiderato. Per i passaggi successivi si utilizzerà solo uno tra i valori di Γ provati, cioè $\Gamma = 20$.

Si completa la taratura dei guadagni robot impostando il valore di τ (Figure 6.10 e 6.11).

L'influenza di questo parametro modifica fortemente il livello di "accondiscendenza" del controllore: piccoli valori di τ (quindi elevati valori di ff) portano a una maggior riduzione del contributo adattativo fornito da $\dot{a}_s$ (che quindi non raggiunge più la quota di mantenimento intorno ai 6N), in

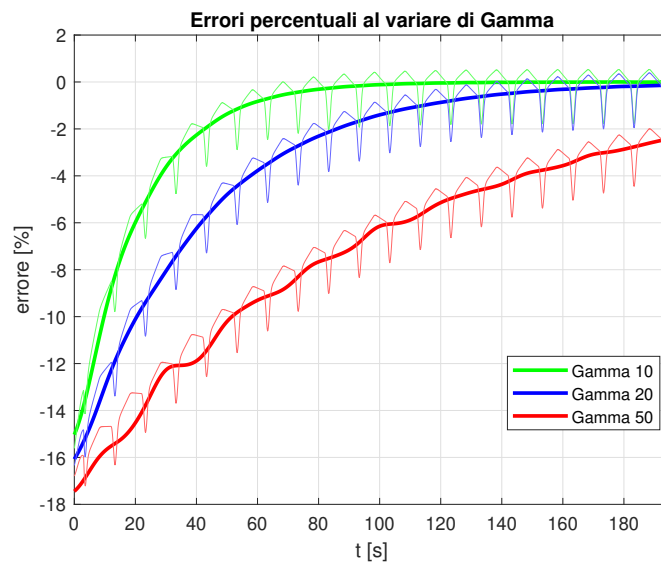


Figura 6.8: Errore di posizione al variare di Γ

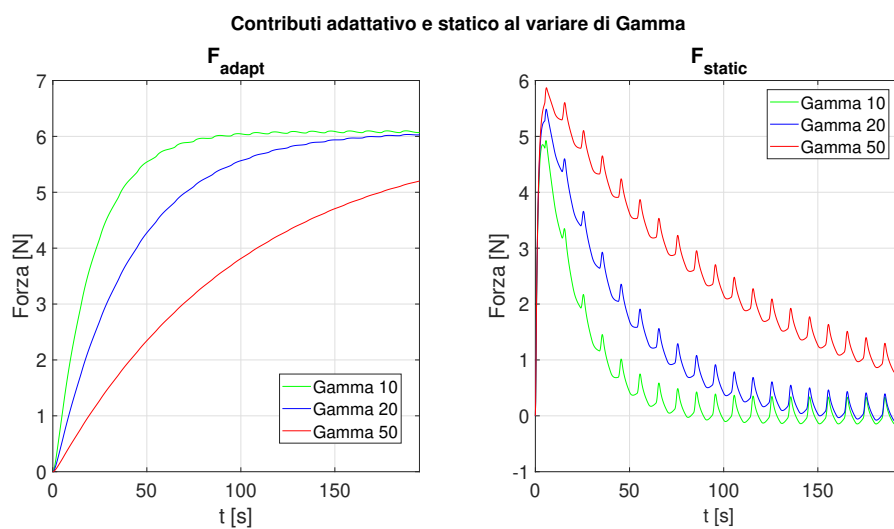


Figura 6.9: Contributi di forza al variare di Γ

quanto con un grande valore di *forgetting* si ha un controllo che dimentica velocemente la dinamica appresa; a ciò consegue quindi un errore a regime possibile più elevato.

In maniera analoga e opposta, elevati valori di τ si ripercuotono in un controllore che mantiene maggiormente il contributo dato dallo *sliding* e quindi che permette effettivamente un errore nel tracking (il tracking perfetto garantito dall'azione del solo Γ non è comunque soddisfatto quando si aggiunge il termine di *forgetting*) ma che sarà senz'altro minore rispetto a quello concesso da costanti di tempo minori.

Volendo anticipare quanto si vedrà una volta inserito il modello del paziente simulato, si può vedere il comportamento del controllore come "severo" per τ alti e "accondiscendente" per τ ridotti.

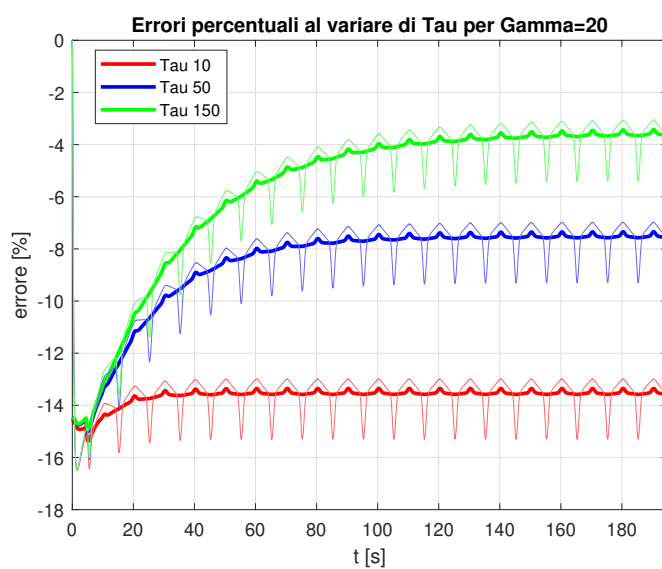


Figura 6.10: Errore di posizione al variare di τ per $\Gamma=20$

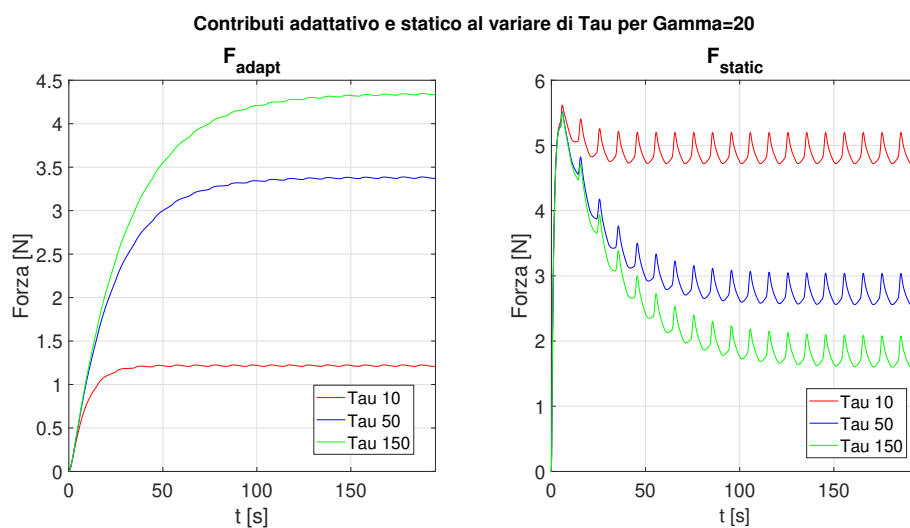


Figura 6.11: Contributi di forza al variare di τ per $\Gamma=20$

- **Taratura guadagni paziente**

Si prosegue impostando i parametri del blocco simulativo "paziente". Per come è stato strutturato il modello, occorre settare il valore dei parametri Kp_{pz} , Kd_{pz} del controllore PD, che definiscono la forza complessiva che il soggetto riesce a fornire durante l'esercizio e il parametro n che invece rappresenta il numero di cicli necessari per l'apprendimento del percorso terapeutico.

Mantenendo costante il valore di $Kd_{pz}=1$ si modifica il guadagno proporzionale da 10 a 400 (Figure 6.12 e 6.13). Questo permette di simulare il comportamento di un paziente "debole", cioè capace di fornire una forza ridotta durante l'esercizio e di un paziente "forte" che al contrario è capace di eguagliare e superare il contributo fornito dal robot. In assenza del fattore di apprendimento si osserva che l'errore e il contributo di forza si mantengono costanti nel tempo a un valore conseguente al Kp_{pz} scelto; inserendo n , invece, si evince come sia errore che forze varino nel tempo, con l'errore che cresce fino a portarsi alla condizione di massima correzione introdotta dal guadagno proporzionale del paziente e le forze che in modo analogo si assestano al valore massimo possibile per il paziente modellizzato (Figure 6.14 e 6.15). In tutti i casi la velocità di apprendimento è definita dal valore di n . Per tutte le simulazioni successive si utilizzerà $Kp_{pz}=400$.

- **Comportamento complessivo robot e paziente**

Si terminano le prove simulate osservando come robot e paziente simulato si comportano durante un ciclo di terapia.

Sono state effettuate prove per due valori di Γ distinti (10 e 20) per i quali è stato modificato τ . Si considera sempre un paziente con buone capacità di movimento ($Kp_{pz}=400$) e apprendimento in 12 cicli.

Occorre in questo caso osservare l'interazione tra i contributi di forza forniti dal robot e dal soggetto, F_{robot} e F_{pz} .

Ricordando che il termine di F_{robot} è dato dalla somma dei contributi statico e adattativo, si può desumere che il suo andamento sia prettamente determinato da quest'ultimo e, in linea con quanto osservato in precedenza, si osserva che il suo contributo ha una diminuzione tanto più repentina quanto

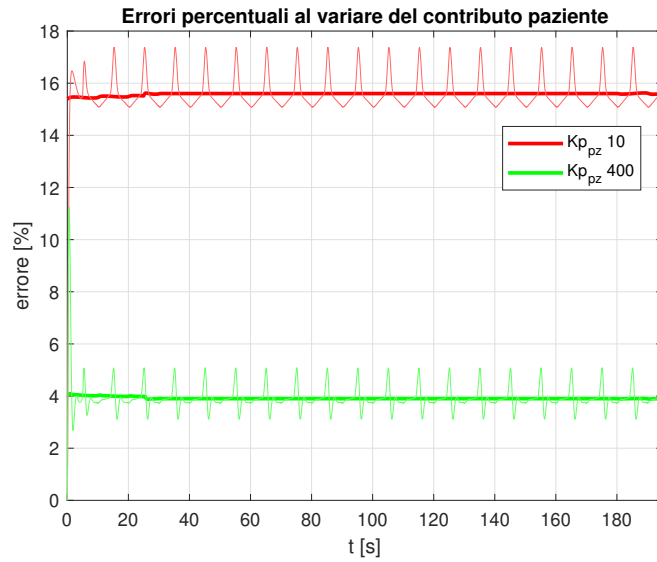


Figura 6.12: Errore di posizione al variare di Kp_{pz}

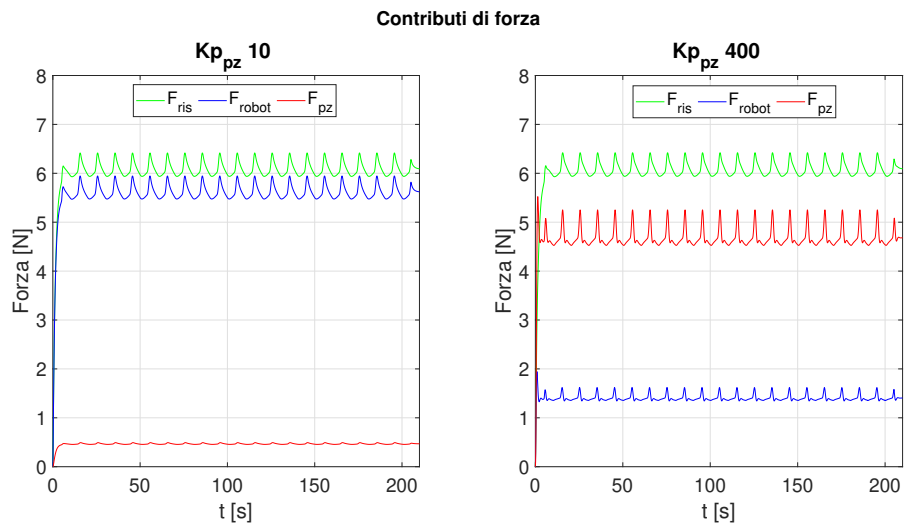


Figura 6.13: Contributi di forza al variare di Kp_{pz}

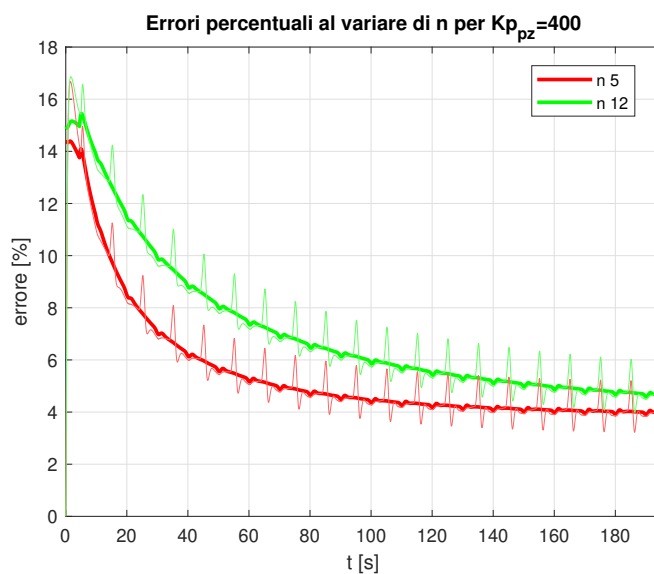


Figura 6.14: Errore al variare di n con $K_{p_{pz}}=400$

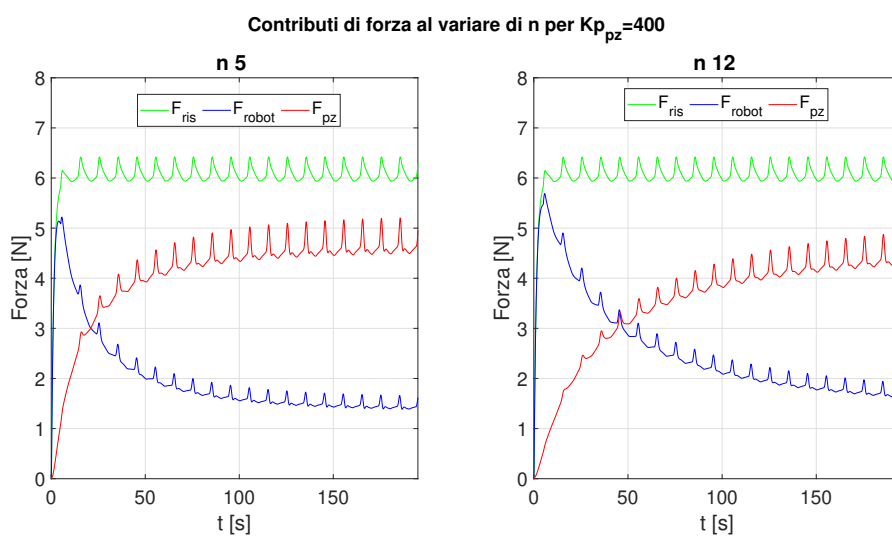


Figura 6.15: Contributi di forza al variare di n con $K_{p_{pz}}=400$

più alto il valore di Γ e basso τ . In questa situazione, infatti, il ff elevato fa sì che il manipolatore sia "accondiscendente", determinando una diminuzione della forza adattativa come conseguenza della rilevazione dello sforzo del paziente, che va a modificare i parametri stimati a . Pertanto, a parità di *adaptive gain* e di F_{pz} , si osserva che un robot "severo" (τ grande) fornirà un'assistenza maggiore e che quindi porterà a un più preciso *tracking*, con minor possibilità del paziente di dare il suo contributo: il controllore è "severo" perchè non permette che il soggetto compia un errore troppo grande. Al contrario per valori di τ piccoli si ha un ff grande e quindi un robot che dimentica velocemente la dinamica del movimento imparato: questo si ripercuote sulla maggior "accondiscendenza" nei confronti del paziente che può subentrare in maniera più o meno evidente al robot, che gli permette di eseguire l'esercizio concedendo un errore di posizione maggiore. Occorre poi evidenziare che lo stesso τ determina la stessa velocità di decadimento al variare di Γ e che l'errore residuo concesso in funzione di τ è comunque minore rispetto a quello che veniva permesso dal solo robot, in quanto comunque l'azione del paziente aiuta nell'inseguimento.

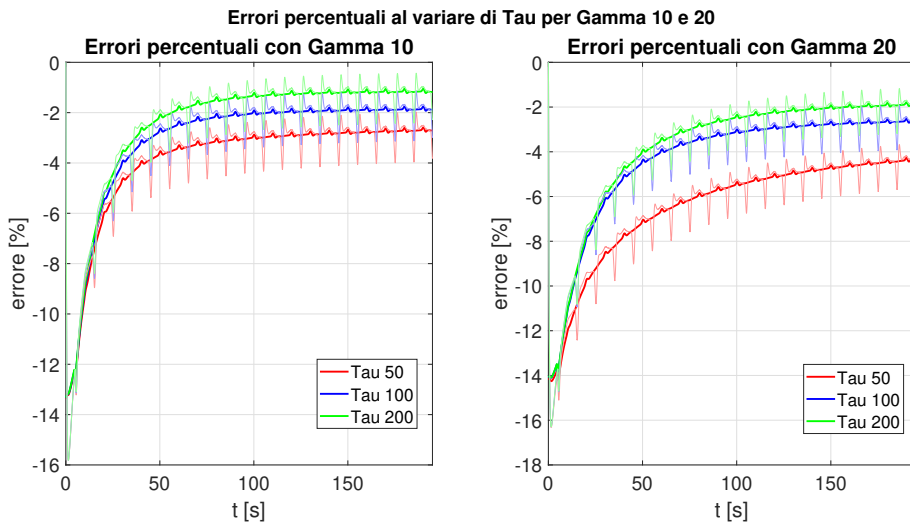


Figura 6.16: Errore al variare di Γ e τ

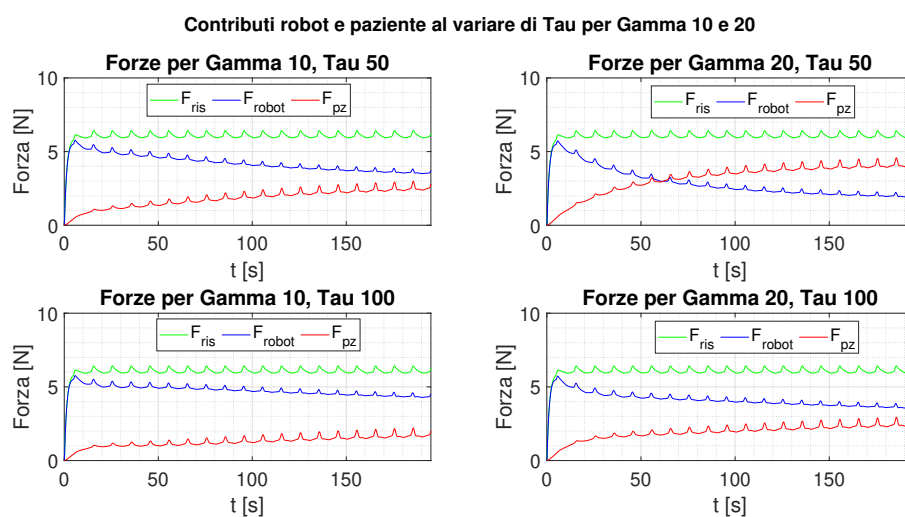


Figura 6.17: Contributi di forza al variare di Γ e τ

6.3 Risultati sperimentali controllo adattativo

In questa sezione verranno riportati i risultati sperimentali ricavati dalle prove effettuate su MariBot 2.0. Tali prove sono state eseguite sul controllo del singolo cavo con massa, dei tre cavi a sostegno dell'ortesi e infine dei giunti azionati dei link.

Ciò che ci si aspetta di osservare è, per le diverse prove, un comportamento che sia in linea con quanto visto nella sezione precedente di simulazioni per il solo caso del singolo cavo: avendo strutturato tutti i controllori secondo la logica riportata nello schema di Figura 6.2, ciò che ci si attende è lo stesso comportamento al variare dei diversi parametri e guadagni.

In tutte le prove eseguite, la procedura di taratura e studio è quella esposta passo passo in 6.2.2, pertanto verrà presentata per intero solo per il singolo cavo, mentre per il controllo dei tre cavi e dei i giunti saranno esposti gli aspetti inerenti ad alcune osservazioni che si sono rese necessarie per il setting dei parametri e infine il comportamento totale dato dalla cooperazione tra robot e paziente.

6.3.1 Risultati sperimentali singolo cavo

Come già visto nella sezione simulativa, anche per le prove reali lo studio del comportamento del controllore inizia con la taratura dei guadagni robot, cui segue quella dei parametri del paziente che portano, alla fine, all'analisi del comportamento totale.

Per cercare di ottenere quanta più corrispondenza i parametri utilizzati sperimentalmente sono identici a quelli utilizzati nelle simulazioni.

- **Taratura guadagni robot**

Si comincia con l'impostare i guadagni del controllore statico, seguendo il già visto criterio basatu sul garantire l'esecuzione dell'esercizio con un errore di tracking residuo intorno al 20%. Questo è garantito per gli attesi valori di $K_p = 120$ e $K_d = 50$. Ancora una volta, la forza risultante si porta ad oscillare nell'intorno del valore corrispondente al peso della massa, con oscillazioni più ampie rispetto allo studio simulativo per motivi inerenti alle non idealità

non modellizzate in simulazione (aspetto comunque già osservato anche per il controllo a basso livello).

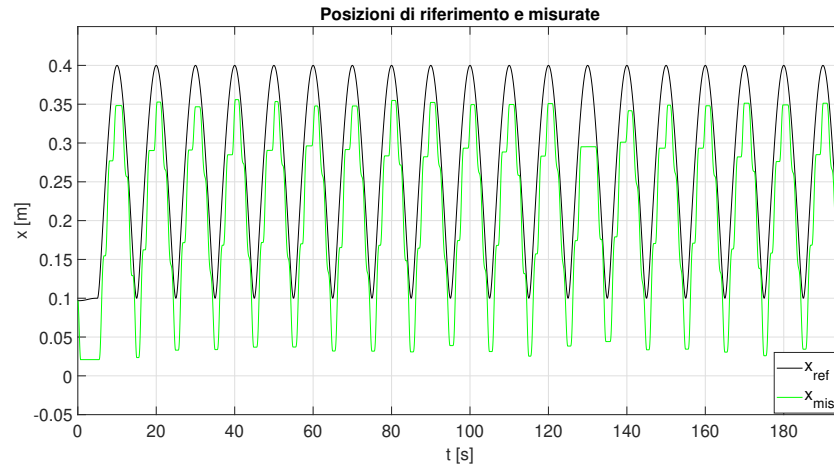


Figura 6.18: Tracking di posizione per controllo statico con $K_p=120$ e $K_d=50$

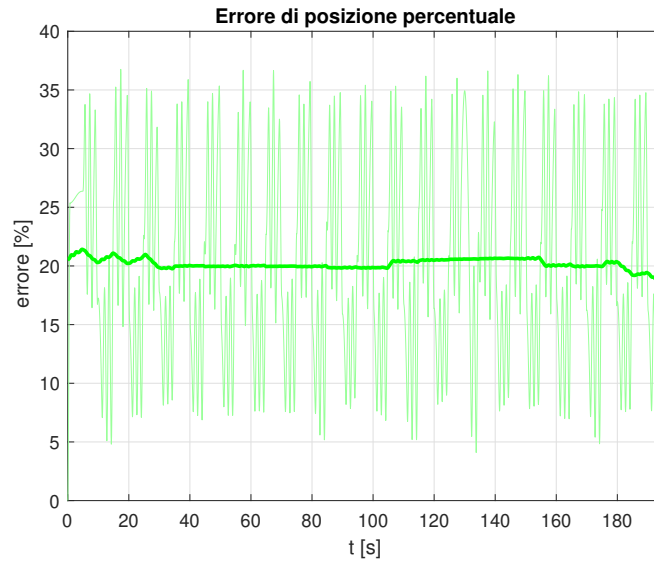


Figura 6.19: Errore di posizione per solo controllo statico con $K_p=120$ e $K_d=50$

Si prosegue con la taratura dei guadagni dell'adattativo, partendo dalla componente di *sliding control*. Occorre sempre precisare che si mantengono $\Lambda = 1.5$ e $\sigma = 0.108$, in linea con quanto visto in precedenza. Ciò che si può aggiungere è che modificando Λ si va a pesare maggiormente l'errore di

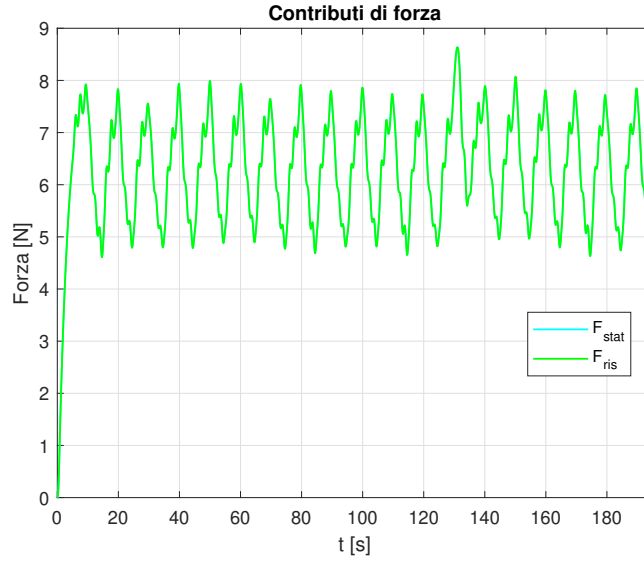
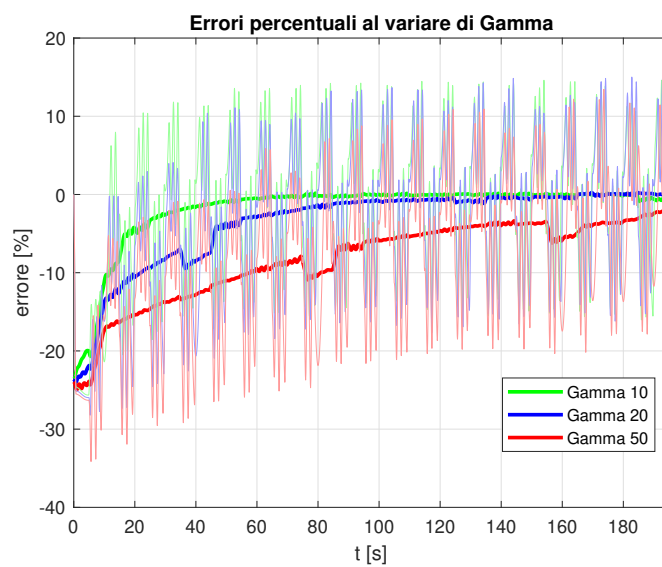
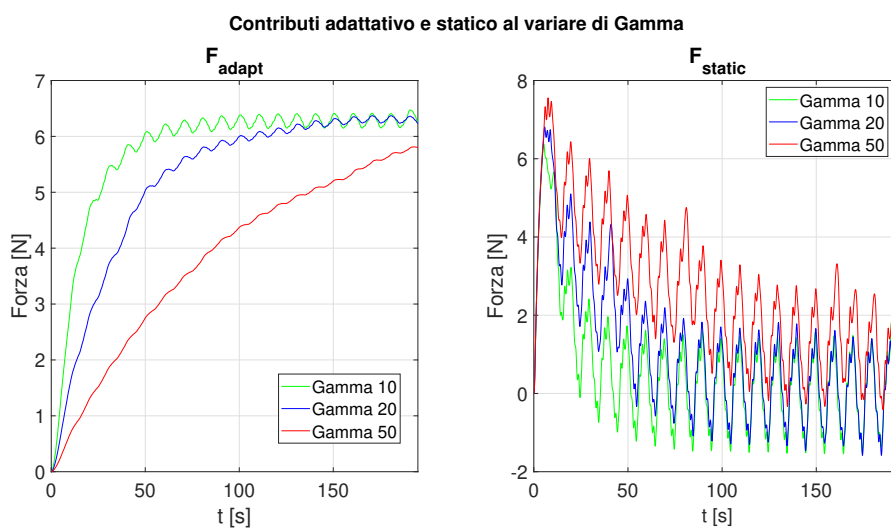


Figura 6.20: Forza risultante per solo controllo statico con $K_p=120$ e $K_d=50$

posizione (se maggiore di uno) o di velocità (se minore di uno) nel calcolo delle *sliding surfaces*: ciò che si può osservare è che valori alti per questo parametro portano a un andamento molto scattoso, in quanto si dà molta più importanza all'errore di posizione piuttosto che alla velocità di esecuzione che, nel caso in esame, risulta comunque non trascurabile.

Si provano a questo punto valori di Γ diversi e si può verificare come in tutti i casi questi determinino l'annullamento dell'errore, con una rapidità inversamente proporzionale al valore del guadagno imposto. Il contributo adattativo è quindi, come atteso, più veloce nel portarsi nell'intorno del valore di mantenimento per Γ ridotti. Si può ovviamente osservare come in maniera duale la F_{static} ha in questi casi una riduzione più rapida ed evidente. Lo studio dei parametri del controllore robot può terminare osservandone il comportamento al variare di τ una volta impostato Γ fisso uguale a 20. Si sta quindi valutando l'azione del contributo di controllo di *forgetting*.

Avendo in mente il concetto di "robot severo per τ grandi" e "robot accondiscendente per τ piccoli" è facile osservare che il comportamento ottenuto sperimentalmente è del tutto in linea con quanto visto nelle simulazioni: il ff riduce il contributo associato alla parte di *sliding control* e al contempo

Figura 6.21: Errore di posizione al variare di Γ Figura 6.22: Contributi di forza al variare di Γ

concede un errore nell'inseguimento della traiettoria; tali aspetti saranno tanto più evidenti quanto più il controllore sarà accondiscendente, dunque per valori di τ piccoli che determinano una maggior velocità nel dimenticare la dinamica appresa. Si può osservare come l'ordine di grandezza di errore e forze sia effettivamente confrontabile con quelli ottenuti in via simulativa.

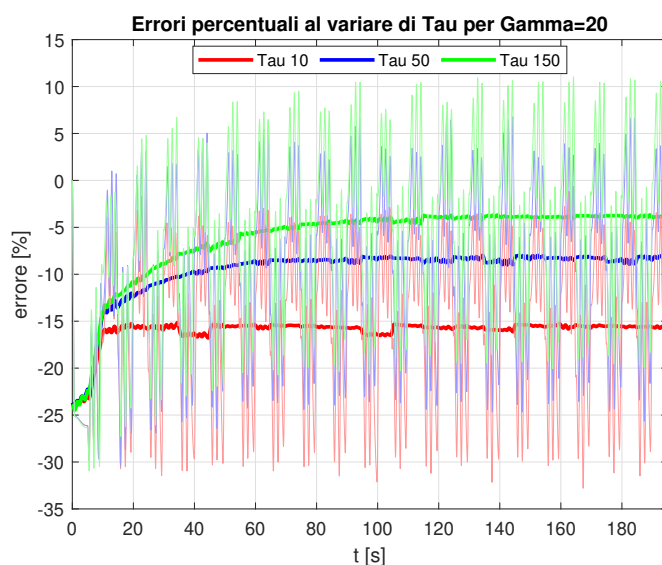


Figura 6.23: Errore di posizione al variare di τ per $\Gamma=20$

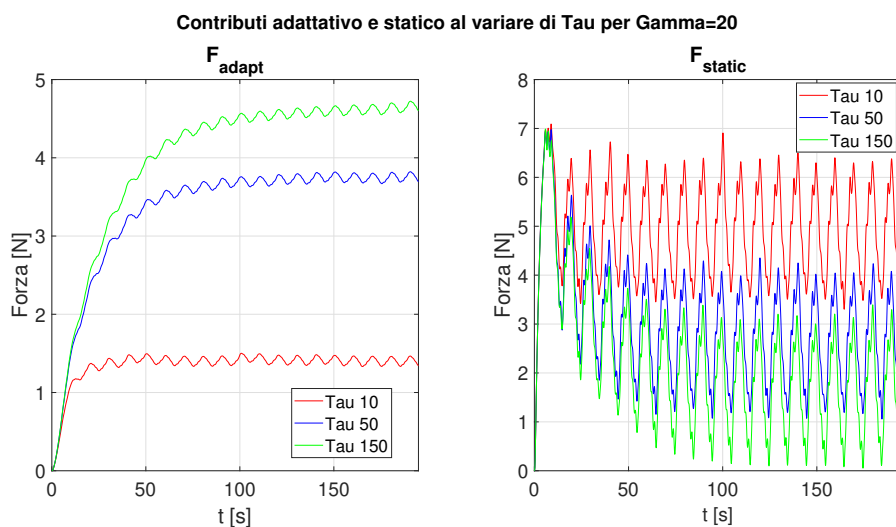


Figura 6.24: Contributi di forza al variare di τ per $\Gamma=20$

- **Taratura guadagni paziente**

Proseguendo nel procedimento di taratura, si passa al settaggio dei parametri del paziente. Si mantiene lo stesso modello impiegato per le simulazioni in quanto in questo modo risulta possibile quantificare con esattezza il contributo del soggetto. I risultati ottenuti sono anche in questo caso corrispondenti a quanto visto nella sezione precedente.

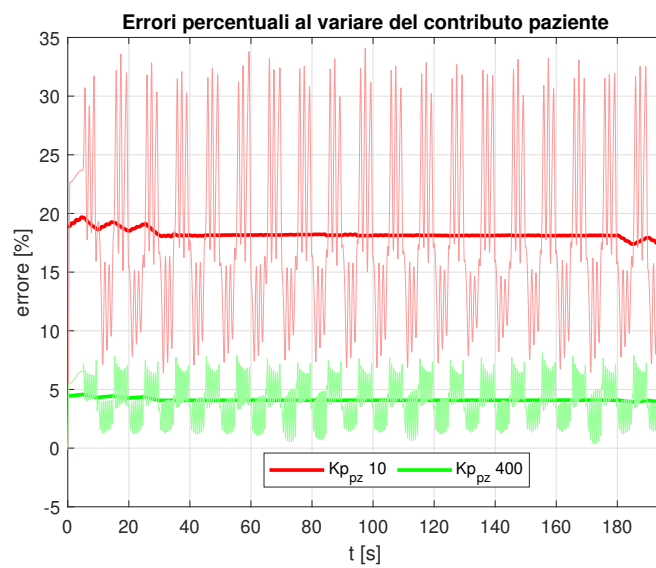


Figura 6.25: Errore di posizione al variare di Kp_{pz}

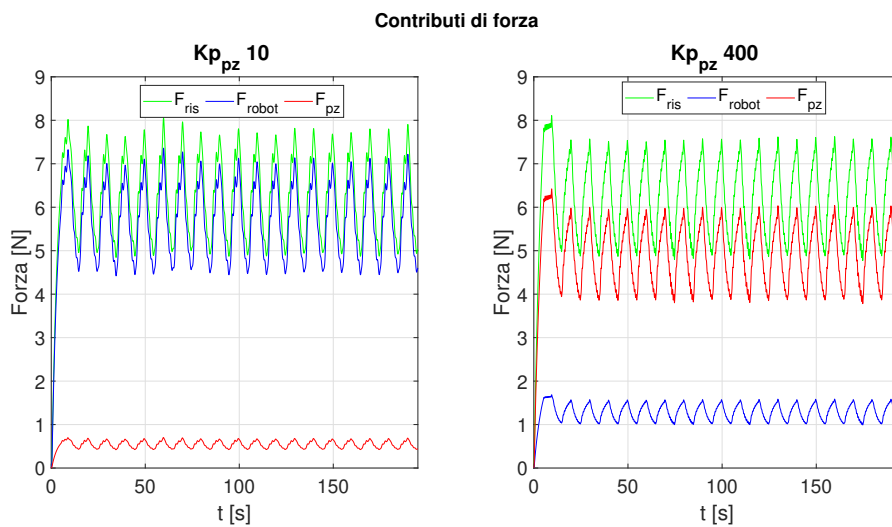


Figura 6.26: Contributi di forza al variare di Kp_{pz}

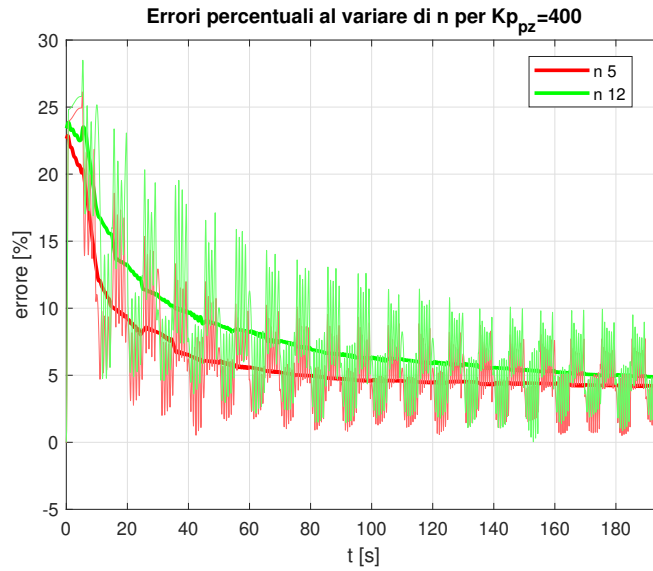


Figura 6.27: Errore al variare di n con $K_{p_{pz}}=400$

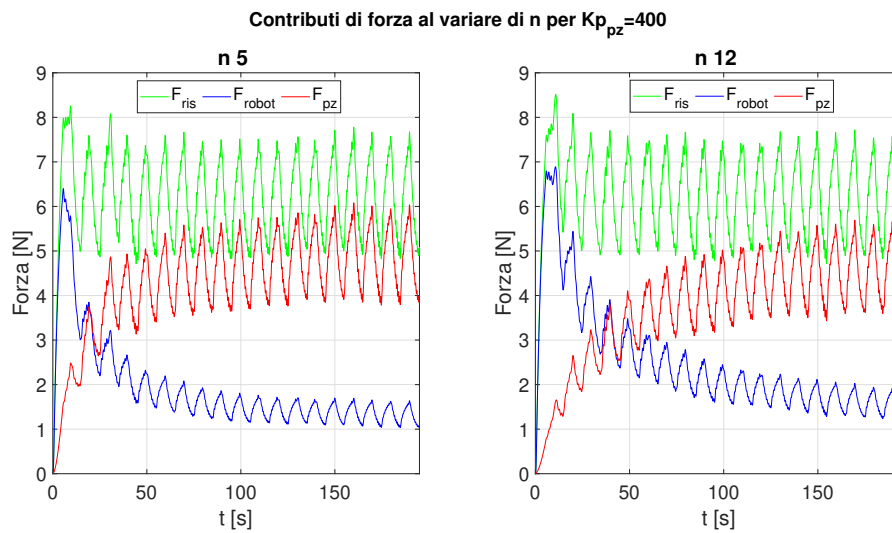


Figura 6.28: Contributi di forza al variare di n con $K_{p_{pz}}=400$

- **Comportamento complessivo robot e paziente**

Si concludono le prove simulative per il singolo cavo analizzando il comportamento complessivo dato dalla cooperazione tra robot e paziente. Anche in questo caso si mantiene un paziente "capace" con guadagno $K_{pz}=400$ e $n=12$ e si modificano i valori di Γ e τ per quanto riguarda il robot.

Dai risultati riportati nelle figure che seguono si può anche in questo caso constatare la corrispondenza con il comportamento simulativo: il robot tenderà a fornire una maggior assistenza per valori ridotti dell'*adaptive gain* e per τ grandi, imponendo un inseguimento più preciso della traiettoria desiderata e una correzione più repentina, limitando l'azione del paziente. Al contrario quest'ultimo sarà più libero di esprimersi quando il contributo di τ risulta ridotto o per uno *sliding control* meno pronto (Γ maggiore). Si può anche osservare come, per crescenti valori di Γ , il punto di inizio dell'andamento dell'errore sia più lontano dallo zero e che comunque rimanga lo stesso per diversi valori di τ a parità dell'*adaptive gain*. Allo stesso modo osservando il comportamento per Γ diversi ma a parità di ff si constata l'assestamento nell'intorno della stessa quota di errore residuo.

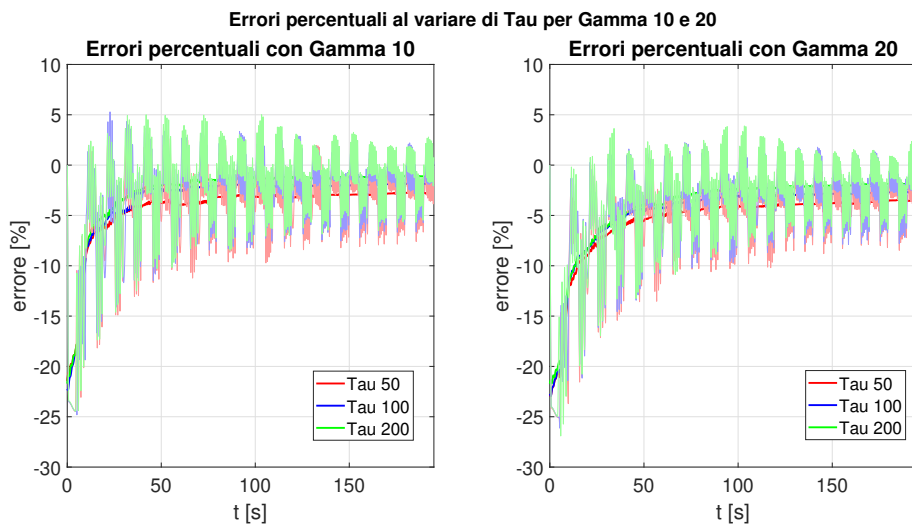


Figura 6.29: Errore al variare di Γ e τ

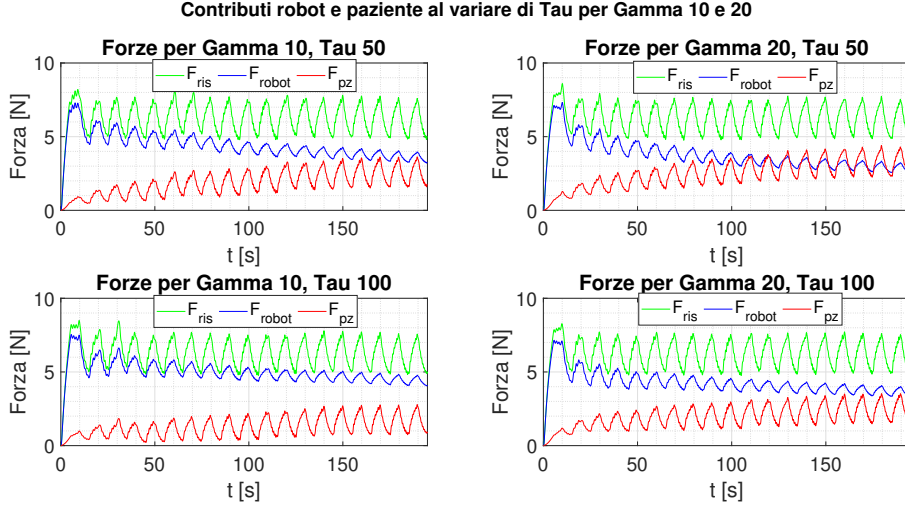


Figura 6.30: Contributi di forza al variare di Γ e τ

6.3.2 Risultati sperimentali tre cavi

Per realizzare il controllo adattativo del sistema completo di sollevamento dell'ortesi si replica su ciascuno dei tre cavi lo schema già impiegato per il singolo. Questo significa che si avranno tre controllori che sfruttano come ingressi gli errori di posizione e velocità del cavo SX, DX e BACK e che forniranno conseguentemente in uscita il contributo di forza necessario per il movimento.

Per quanto concerne il settaggio e la struttura dei diversi parametri, invece, occorre fare una serie di precisazioni.

Innanzitutto occorre precisare che si suppone che gli spazi di lavoro dei tre cavi siano tra loro uguali e ugualmente discretizzati, pertanto il numero p di RBF, il posizionamento dei centroidi μ e il relativo valore di σ viene mantenuto uguale nei tre schemi. I valori saranno gli stessi già impiegati per il singolo cavo, che portano dunque ad ottenere un $\sigma=0.108$. Anche Λ viene mantenuto a 1.5.

Entrando poi nel dettaglio dei parametri che direttamente influenzano la risposta del controllore, cioè i guadagni del PD Kp , Kd e quelli dell'adattativo Γ e τ , si è reso necessario distinguere tra controllo dei cavi anteriori (SX, DX) e posteriore (BACK). La motivazione che ha spinto a tale scelta è da ricercarsi nello stesso motivo già esposto per il controllo a basso livello, cioè che il singolo cavo BACK deve essere in grado, da solo, di fornire la stessa forza dei cavi anteriori per con-

sentire il movimento della ortesi.

A questo scopo, dunque, si vedrà differenziato il settaggio dei guadagni proporzionali e derivativo in maniera tale garantire la stessa quota di correzione dell'errore di tracking e allo stesso modo si ragionerà per l'*adaptive gain*, che sarà scelto secondo lo stesso criterio dei guadagni del PD statico cui si aggiunge la necessità di avere lo stesso contributo tra F_{static} e F_{adapt} . Se così non fosse infatti, si otterrebbe un controllore sbilanciato tra cavi anteriori e posteriori.

Si evidenzia ancora una volta come la procedura seguita nell'eseguire le prove sia di fatto la medesima già esposta per le simulazioni e prove sperimentali del singolo cavo; per brevità si riportano quindi solo gli aspetti inerenti alle osservazioni appena descritte.

- **Taratura parametri robot**

Le prime prove sono state eseguite tarando i tre controllori PD statici e adattativi nella stessa maniera. In figura 6.31 sono riportati gli andamenti per il setting dei $Kp_{SX}=Kp_{DX}=Kp_{BACK}=50$ (con guadagno derivativo sempre definito come un centesimo del proporzionale) e di $\Gamma_{SX}=\Gamma_{DX}=\Gamma_{BACK}=20$ e con $\tau=50$. Per brevità si chiamerà questa taratura "taratura uniforme". Ciò che si può osservare è quindi la netta differenza di errore tra cavi anteriori e posteriori, con il BACK che non è in grado di apportare una correzione efficace come quella di SX e DX; si osserva inoltre come, a causa del guadagno proporzionale insufficiente, risulti ridotto anche il contributo fornito dal controllore adattativo. Ciò evidentemente sfocia in una discrepanza tra il rapporto tra contributi adattativo e statico (F_{adapt}/F_{static}) tra i cavi anteriori (per i quali è confrontabile) e posteriore (nei quali è invece minore).

Appresa quindi la necessità di differenziare i guadagni ("taratura differenziata"), si modifica inizialmente il valore dei contributi proporzionali così che sia garantito un errore di posizione nell'intorno del solito 20% per tutti e tre i cavi: tale obiettivo è raggiunto imponendo $Kp_{SX}=Kp_{DX}=50$ e $Kp_{BACK}=100$.

Evidentemente l'aver scelto parametri diversi nel controllore statico si ri-

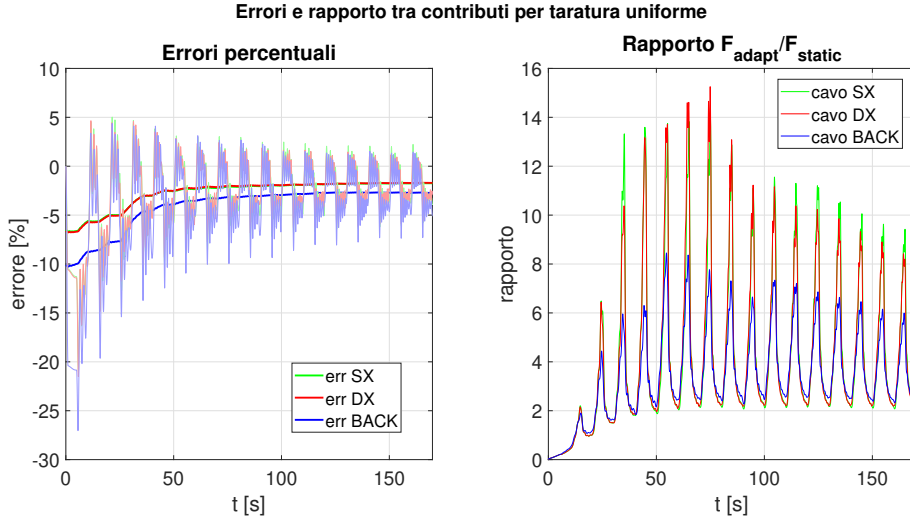


Figura 6.31: Errore percentuale (sinistra) e rapporto tra contributi (destra) per taratura uniforme

percuote anche nella necessità di scegliere diversamente anche il parametro Γ , differenziandolo tra quello adibito ai cavi SX e DX e quello dedicato al solo BACK. Il criterio impiegato per la scelta, eseguito tramite procedura di *trial and error* nel caso di solo contributo di *sliding control*, si basava sull'osservare due caratteristiche nel comportamento risultante: l'errore di posizione e il rapporto tra i contributi statico e adattativo.

Per ottenere un controllore bilanciato, infatti, è necessario in primis che l'inseguimento di posizione sia omogeneo per ogni cavo e, in secondo luogo, che vi sia per ogni caso la medesima proporzione tra F_{static} e F_{adapt} . Ciò che va osservato in questo secondo punto è, nel dettaglio, non tanto il valore di regime attorno cui si portano i diversi contributi, che risulterà per ogni valore dell'*adaptive gain* uguale al valore del peso corrispondente sentito da quel determinato motore (già osservato per il controllo a basso livello), quanto più l'ampiezza delle oscillazioni attorno a tale valore. Si osserva infatti che al ridursi di Γ si hanno ampiezze maggiori per il contributo adattativo e conseguentemente oscillazioni dello statico ridotte.

A valle di svariati tentativi è stato possibile costruire la seguente relazione:

$$\Gamma_{anteriore} = 2 \cdot \Gamma_{posteriore} \quad (6.9)$$

Quindi ciò che si evince è che un guadagno proporzionale statico maggiore richiede, per il mantenimento dell'errore e del rapporto adattativo/statico, anche un maggior sforzo proprio da parte della componente di *sliding*, ottenuto quindi per valori ridotti di Γ .

Di seguito sono riportati, al variare di Γ , gli andamenti degli errori percentuali di posizione, che risultano da una parte confrontabili tra cavi anteriori e posteriore, come desiderato, dall'altra capaci di replicare il comportamento noto dalle prove con il singolo cavo di portarsi con rapidità crescente al decrescere dell'*adaptive gain*. Vengono anche rappresentate F_{adapt} e F_{static} nelle medesime condizioni e un grafico che definisce il rapporto medio tra i contributi.

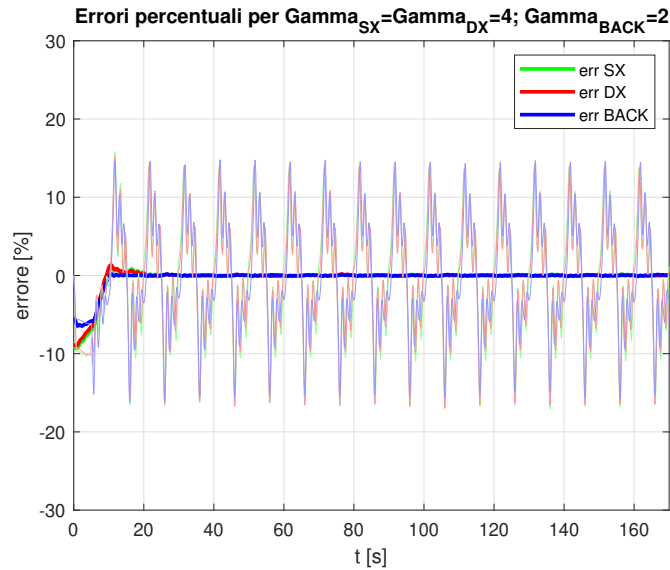


Figura 6.32: Errore di posizione per $\Gamma_{SX} = \Gamma_{DX} = 4$; $\Gamma_{BACK} = 2$

Avendo effettuato la taratura in questo modo, cioè bilanciando per tutti i casi le quote di forza prodotte dalla componente adattativa e da quella statica, è stato possibile implementare la parte di *forgetting* definendo, sia per i cavi anteriori che per il posteriore, lo stesso valore di τ . Si è osservato, infatti, come la riduzione del contributo adattativo e l'errore residuo concesso fossero sempre dello stesso ordine di grandezza per i tre cavi, a parità

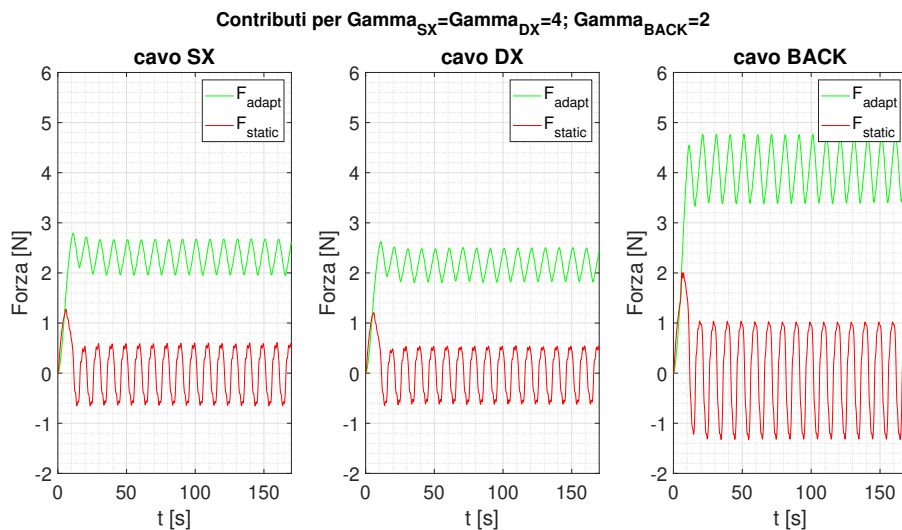


Figura 6.33: Contributi di forza per $\Gamma_{SX}=\Gamma_{DX}=4$; $\Gamma_{BACK}=2$

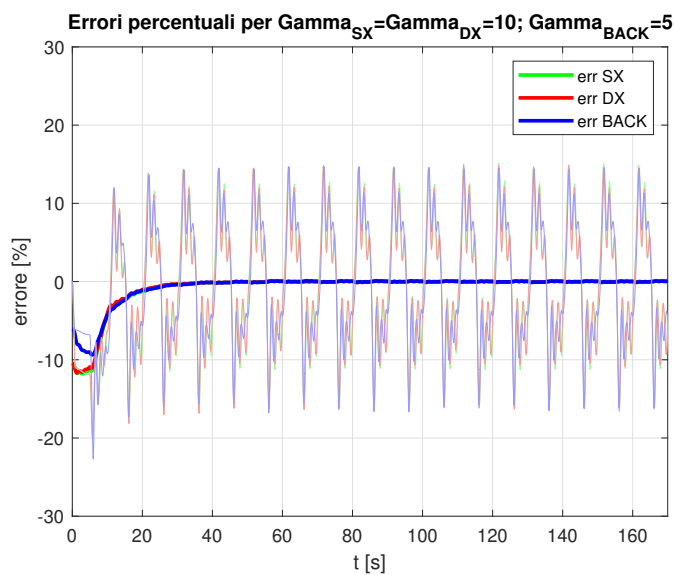


Figura 6.34: Errore di posizione per $\Gamma_{SX}=\Gamma_{DX}=10$; $\Gamma_{BACK}=5$

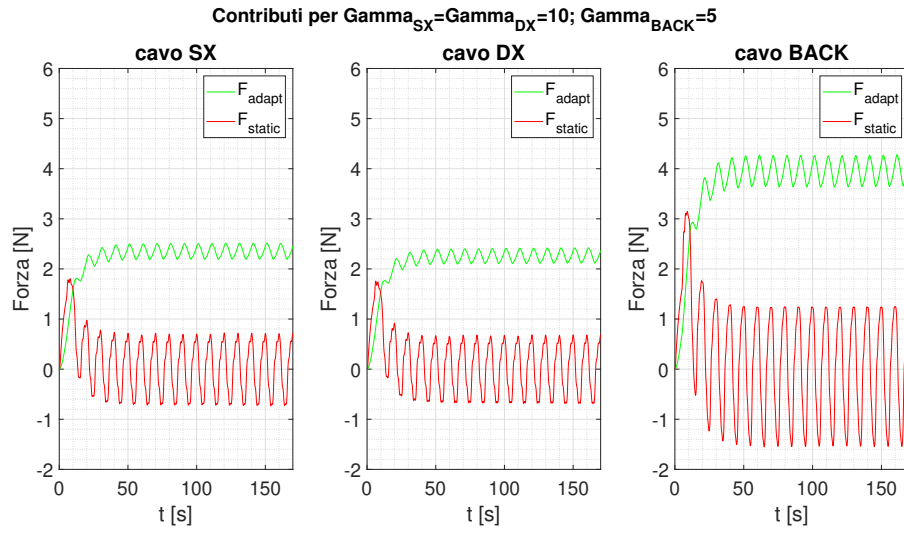


Figura 6.35: Contributi di forza per $\Gamma_{SX} = \Gamma_{DX} = 10$; $\Gamma_{BACK} = 5$

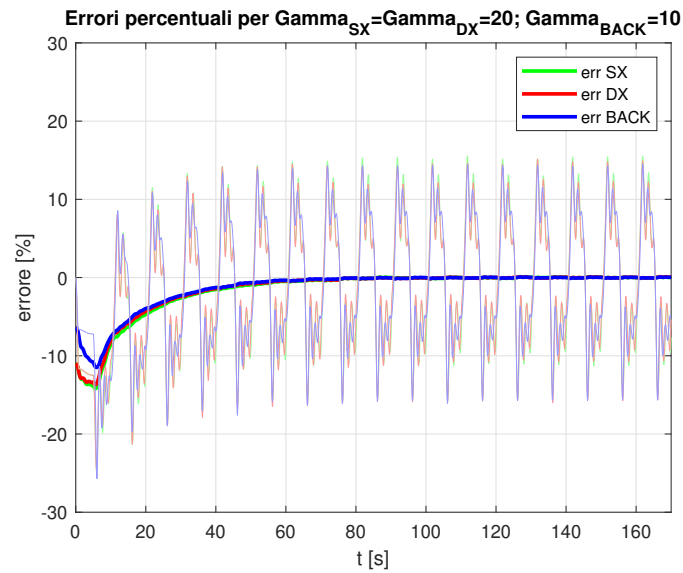


Figura 6.36: Errore di posizione per $\Gamma_{SX} = \Gamma_{DX} = 20$; $\Gamma_{BACK} = 10$

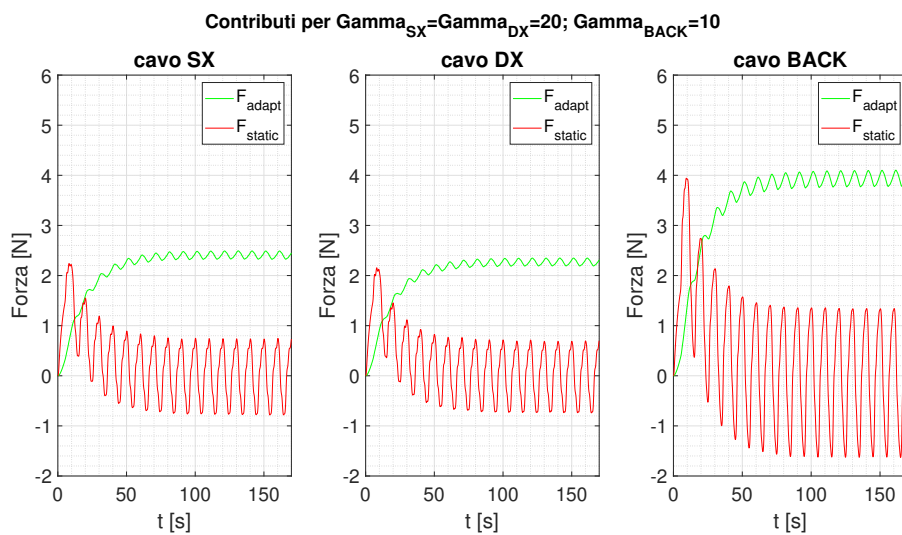


Figura 6.37: Contributi di forza per $\Gamma_{SX} = \Gamma_{DX}=20$; $\Gamma_{BACK}= 10$

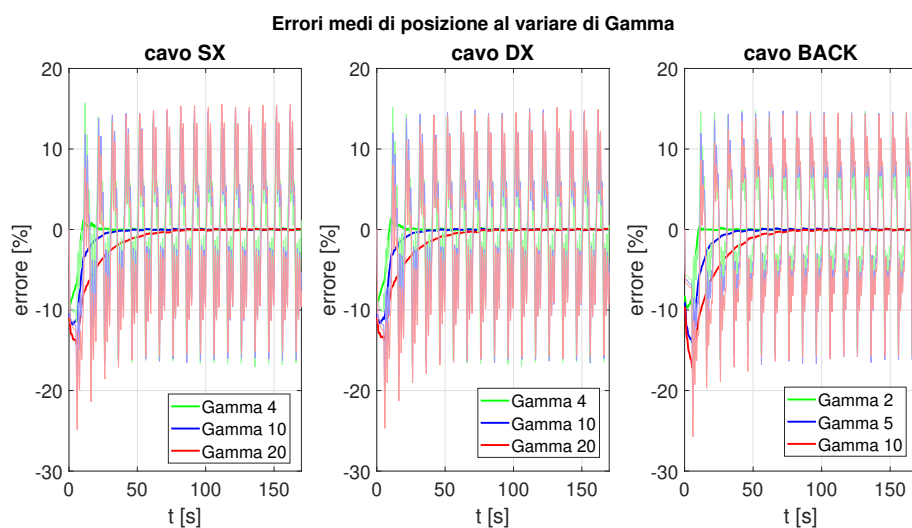


Figura 6.38: Errore di posizione per $\Gamma_{SX} = \Gamma_{DX}=20$; $\Gamma_{BACK}= 10$

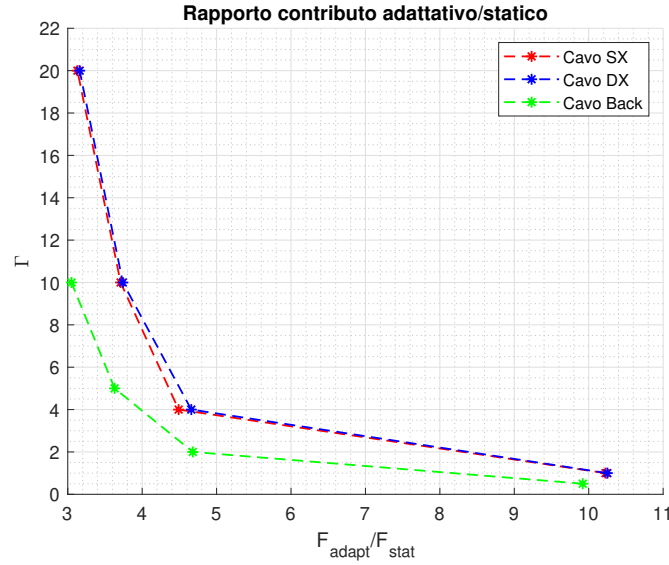


Figura 6.39: Rapporto tra contributi F_{adapt}/F_{stat} al variare di Γ

di ff imposto. Nei casi testati sono stati utilizzati valori di gamma pari a 1 per cavo SX e DX e di 0.5 per il BACK.

- **Taratura guadagni paziente**

Per gli stessi ragionamenti fatti per il controllore proporzionale, è stato modellizzato il paziente differenziando il guadagno proporzionale tra anteriore e posteriore. Come per il solo robot, questo si è reso necessario affinché cavi SX, DX e BACK forniscano lo stesso contributo di forza paziente rispetto a quello dato dal manipolatore, correggendo conseguentemente la stessa parte di errore. Per quanto riguarda la modellizzazione della costante temporale τ_{pz} non ci sono state problematiche di adattamento, in dipendente solo dal numero di cicli di esercizio.

Per le prove è quindi stato definito un paziente con buone capacità di movimento, simulato con guadagni $K_{pzSX}=K_{pzDX}=300$ e $K_{pzBACK}=600$. In tutti i casi il guadagno derivativo rimane un centesimo del proporzionale. Per quanto concerne l'apprendimento, invece, sono stati mantenuti gli $n=12$ cicli di terapia.

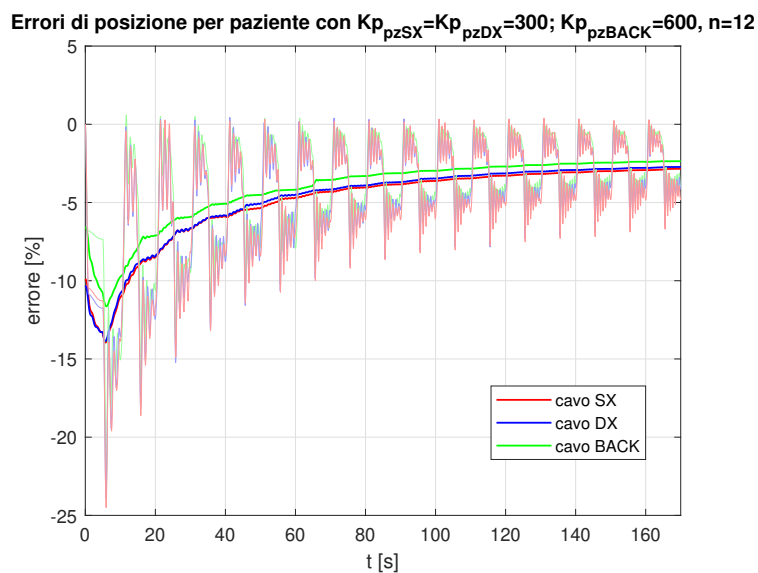


Figura 6.40: Errore di posizione per $K_{p_{pzSX}} = K_{p_{pzDX}} = 300$; $K_{p_{pzBACK}} = 600$ e $n=12$

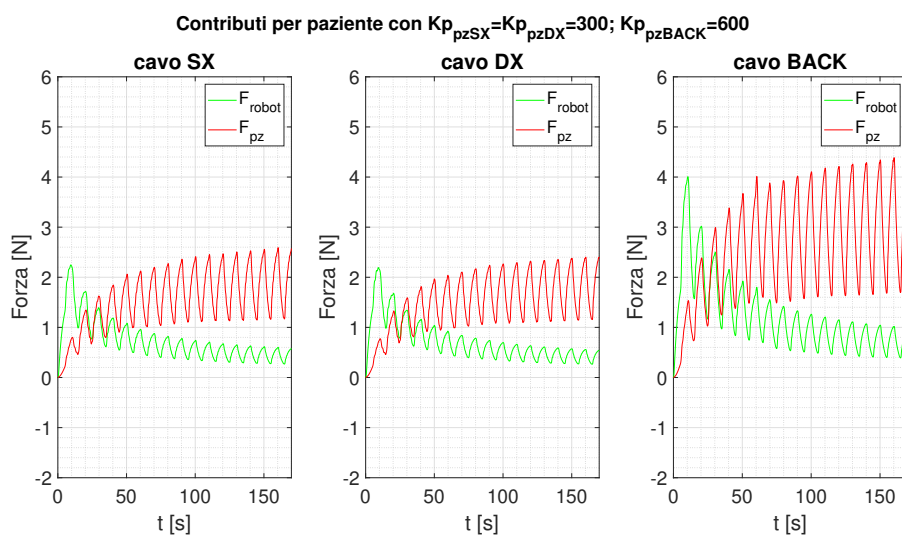


Figura 6.41: Contributi di forza per $K_{p_{pzSX}} = K_{p_{pzDX}} = 300$; $K_{p_{pzBACK}} = 600$ e $n = 12$

- **Comportamento complessivo robot e paziente**

Le modifiche e osservazioni apportate al controllore e descritte nei punti precedenti fanno sì che alla fine si riescano ad ottenere le interazioni tra robot e paziente attese e già osservate nel caso del singolo cavo.

Testando il comportamento del controllore per due diversi set di Γ e due valori di τ si osserva innanzitutto l'effettivo mantenimento delle proporzioni tra contributi robot e paziente e dell'errore, in secondo luogo sono verificati tutti i punti chiave della logica adattativa, ossia il fatto l'errore cala in maniera più rapida al diminuire di Γ o al crescere di τ (controllore più pronto e "severo") e la differenziazione dell'assistenza fornita dal controllore che permette al paziente di fornire un contributo più o meno elevato in funzione di Γ e τ . Si riconoscono anche i comportamenti già osservati nel singolo cavo inerenti alle quote di errore di partenza e finale residuo al variare di Γ e a parità di τ e viceversa.

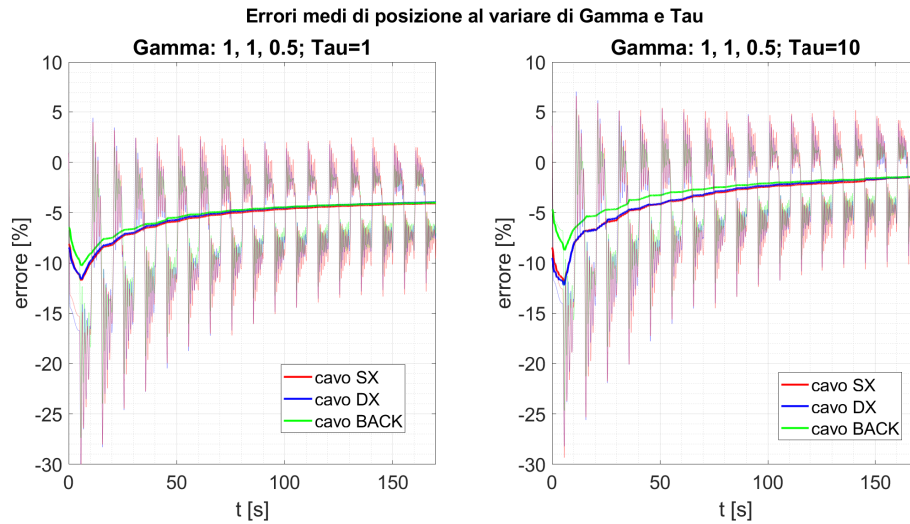


Figura 6.42: Errore di posizione per $\Gamma_{pzSX} = \Gamma_{pzDX} = 1$; $\Gamma_{pzBACK} = 0.5$ al variare di τ

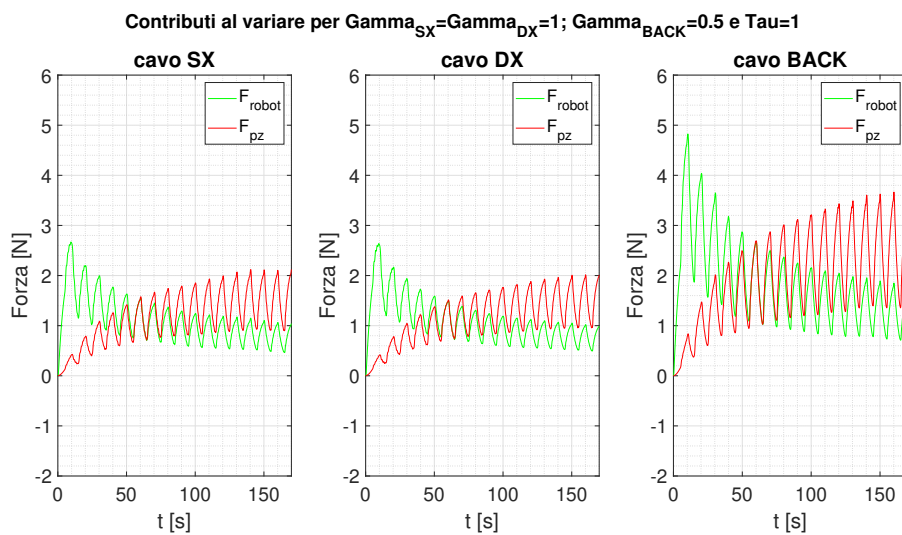


Figura 6.43: Contributi di forza per $\Gamma_{pzSX} = \Gamma_{pzDX} = 1$; $\Gamma_{pzBACK} = 0.5$ e $\tau=1$

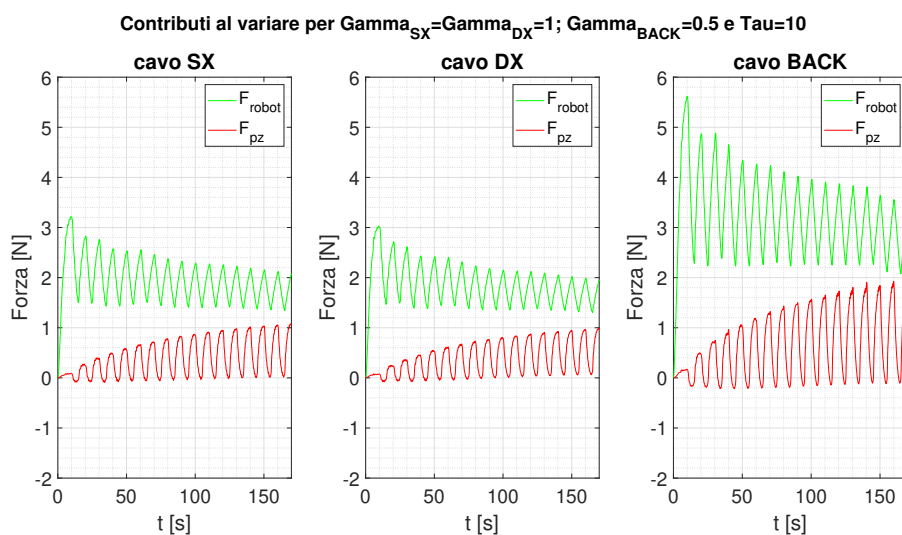


Figura 6.44: Contributi di forza per $\Gamma_{pzSX} = \Gamma_{pzDX} = 1$; $\Gamma_{pzBACK} = 0.5$ e $\tau=10$

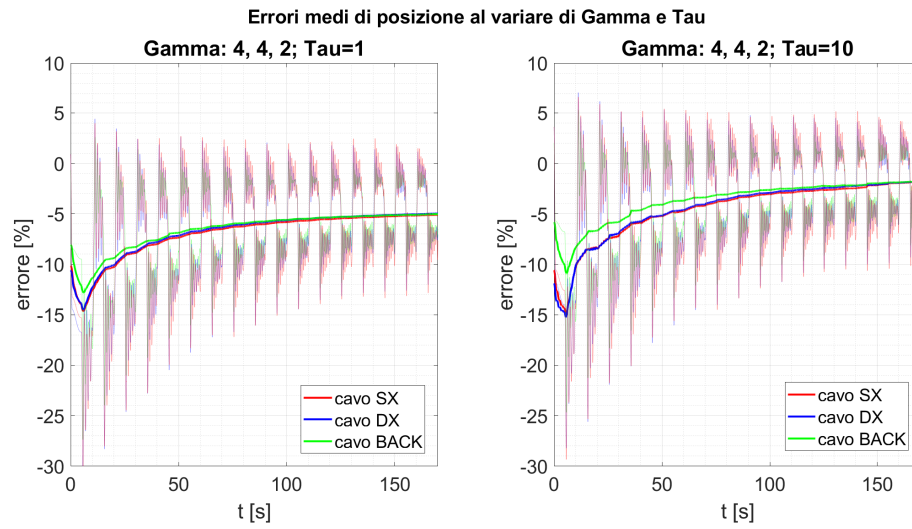


Figura 6.45: Errore di posizione per $\Gamma_{pzSX} = \Gamma_{pzDX} = 4$; $\Gamma_{pzBACK} = 2$ al variare di τ

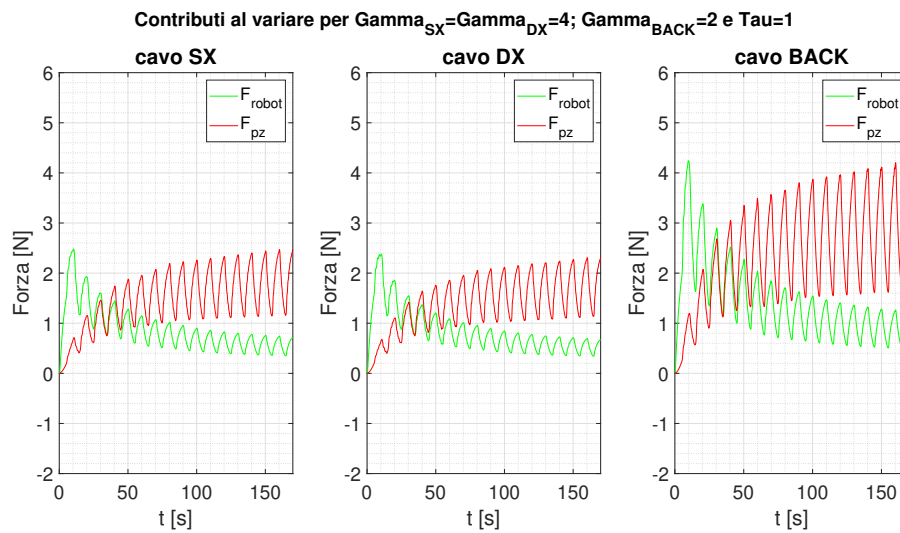


Figura 6.46: Contributi di forza per $\Gamma_{pzSX} = \Gamma_{pzDX} = 4$; $\Gamma_{pzBACK} = 2$ e $\tau=1$

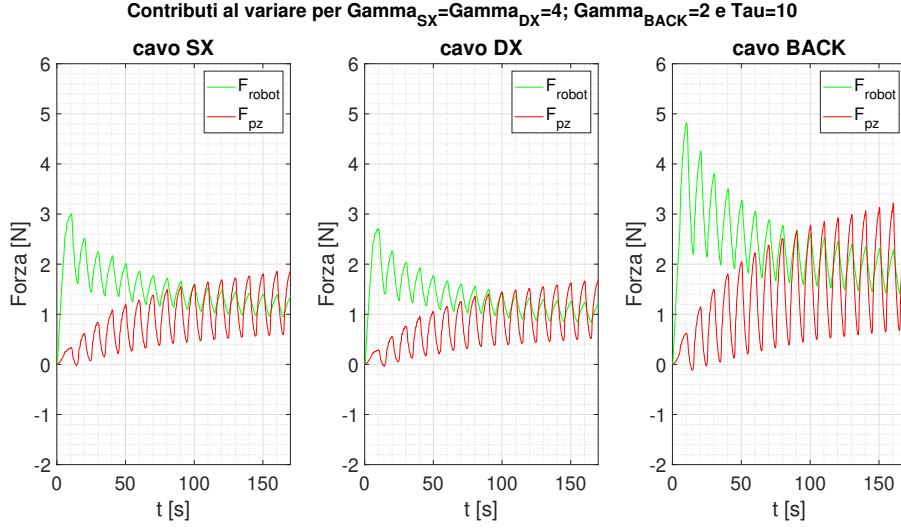


Figura 6.47: Contributi di forza per $\Gamma_{pzSX} = \Gamma_{pzDX} = 4$; $\Gamma_{pzBACK} = 2$ e $\tau=10$

6.3.3 Risultati sperimentali giunti

Si riportano ora le prove sperimentali eseguite applicando il controllo adattativo ai giunti del manipolatore.

Anche in questo caso lo schema di controllo adottato è un adattamento di quello generale 6.2 con la distinzione tra il controllo del giunto 1 e il controllo del giunto 3. Un'ulteriore differenza risiede nell'uscita di tali schemi che, nei casi precedentemente esposti dei cavi, era rappresentata da una forzante in [N], mentre ora definisce la coppia ai giunti 1 e 3 in [Nm] necessaria per quel determinato motore per l'esecuzione della traiettoria desiderata. Occorre precisare, inoltre, che tale riferimento di posizione è fornito in base agli angoli di LINK1 e LINK2, per motivi legati alla maggior semplicità nel riconoscere nella realtà gli spostamenti e i vincoli spaziali da imporre, che altrimenti sarebbero risultati più complessi da individuare se riferiti al LINK3, a causa della catena cinematica costituita dal parallelogramma.

Questa, inoltre, fa sì che gli spazi di lavoro per i due membri siano distinti, anche in funzione della configurazione scelta per il braccio: in tutti i casi, comunque, si avrà un range di rotazione per il LINK1 doppio rispetto a quello concesso al LINK2, con 280° di range per il primo e 140° per il secondo. Col fine di ottenere un controllo adattativo equilibrato, quindi, si è reso necessario discretizzare nello

stesso modo lo spazi di lavoro dei due link; ciò è stato possibile modificando il numero p di RBF equispaziate all'interno del range di interesse: tale aspetto verrà analizzato successivamente.

Sempre nell'ottica di creare una logica di controllo bilanciata, poi, con un ragionamento del tutto simile a quello esposto per le prove con tre cavi, è stato adottato l'approccio della "taratura differenziata" tra il LINK1 e LINK2 (per chiarezza si intenderà sempre il controllo riferito ai link, anche se, come detto pocanzi, di fatto le coppie create sono quelle ai giunti 1 e 3). Se per i tre cavi tale scelta era stata dettata dal fatto che il cavo BACK richiedeva uno sforzo statico maggiore in quanto da solo questo deve garantire le forzanti fornite da SX e DX nella parte anteriore, nel caso del controllo dei giunti questa decisione è stata determinata più che altro per aspetti di natura inerziale e cinematica. Eseguendo un ciclo di esercizio, infatti, si è potuto osservare come il movimento del LINK1 influenzi pesantemente la precisione nella rotazione del LINK2, ad esso collegato e dipendente per le relazioni cinematiche precedentemente esposte. Tali fenomeni risultano tanto più evidenti quanto più è elevata la velocità di esecuzione desiderata. Per garantire, quindi, che entrambi i link riescano a correggere nello stesso modo il movimento (garantendo il medesimo errore percentuale) si è imposto un valore di guadagno proporzionale maggiore per il secondo link.

Conseguentemente a tale imposizione, come già esposto per il controllo dei tre cavi con ortesi, si è dovuto quindi distinguere anche il valore dell'*adaptive gain*, ricercando una proporzionalità tra Γ_1 e Γ_2 che garantisca lo stesso rapporto tra forza adattativa e statica, oltre che di correzione dell'errore.

Un'ultima osservazione interessante riguarda il valore di Λ impiegato. Legato all'alta velocità desiderata per il ciclo di riferimento, abbinata alle elevate masse e inerzie in gioco, è stato osservato come un valore alto per tale parametro portasse ad un andamento molto scattoso del braccio, con oscillazioni sempre più ampie fino a portarsi a condizioni di instabilità. Per risolvere tale problema, dunque, si è impiegato un $\Lambda=0.25$ che pesa maggiormente l'errore di velocità rispetto a quello di posizione nel calcolo delle *sliding function*. Ciò ha permesso di ottenere un andamento decisamente più dolce e, al contempo, un buon inseguimento di posizione.

Dal momento che il procedimento seguito e i risultati parziali seguito sono del tutto analoghi e confrontabili con quelli descritti per il controllo dei tre cavi, si riportano in maniera più riassuntiva i vari punti, fino al raggiungimento delle prove finali date dall'interazione del robot e paziente.

- **Taratura parametri robot**

Secondo la logica della "taratura differenziata", i guadagni del controllore PD statico scelti sono: $Kp_1=0.33$ e $Kp_2=0.66$, con guadagno derivativo pari a un centesimo del proporzionale. Ciò ha permesso di avere un errore residuo intorno al 30% per l'interazione del paziente.

Impostando i parametri della parte adattativa, poi, consci delle osservazioni inerenti alla necessità di discretizzazione e velocità descritte nel paragrafo precedente, sono stati utilizzati: $\Lambda=0.25$ (comune ad entrambi i controllori), $p_1=60$ e $p_2=30$. La necessità di scegliere un numero di RBF doppio tra controllore del LINK1 e del LINK2 è dettata dal fatto che esiste la medesima relazione tra gli spazi di lavoro del singolo link: per discretizzare un range doppio, ci sarà bisogno di utilizzare il doppio dei centroidi. Questo ha portato, dunque, a individuare per i due controllori il medesimo valore del parametro di deviazione standard σ , sempre definito secondo la relazione suggerita da Bobrow $\sigma = \alpha \cdot \Delta x$, dove $\alpha \in [0.75, 1]$ e Δx è la distanza tra un centroide e l'altro. Mantenendo $\alpha=1$, si ottengono quindi (riportando gli spazi di lavoro in radianti): $\sigma_1 = 1 \cdot (280 \cdot \pi)/p_1 \cdot 180 = 0.0814$ e $\sigma_2 = 1 \cdot (140 \cdot \pi)/p_2 \cdot 180 = 0.0814$

Per la scelta dei parametri Γ_1 e Γ_2 , attraverso un processo di *trial and error*, è stato possibile individuare la relazione tra i due capace di garantire i requisiti cercati; essa è:

$$\Gamma_1 = 4 \cdot \Gamma_2 \quad (6.10)$$

Ancora una volta si può osservare come insieme alla richiesta di un maggior contributo statico sia sempre necessario abbinarvi un maggior contributo adattativo, ottenuto quindi per valori ridotti di Γ . Ancora una volta si

riportano in grafico il rapporto tra i contributi C_{adapt} su C_{static} nei vari casi, ad evidenziare l'uguaglianza della proporzione cercata.

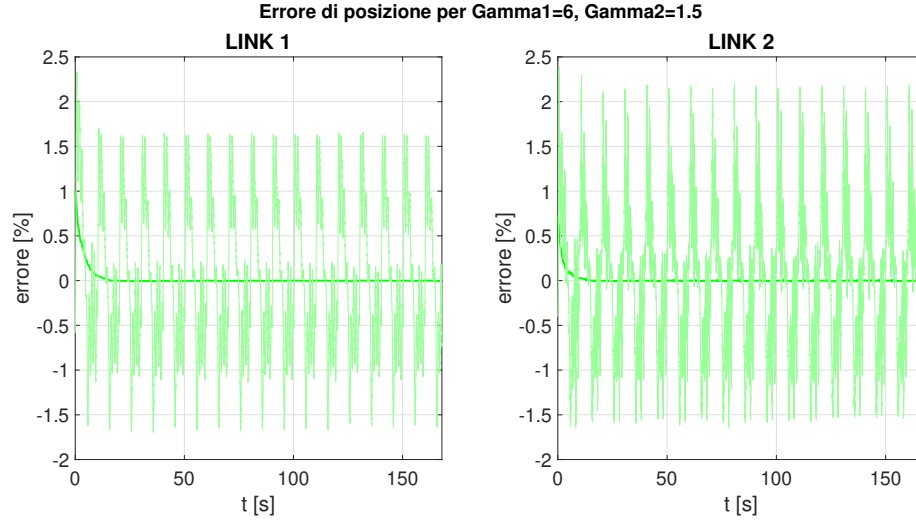


Figura 6.48: Errore di posizione per $\Gamma_1=6$; $\Gamma_2= 1.5$

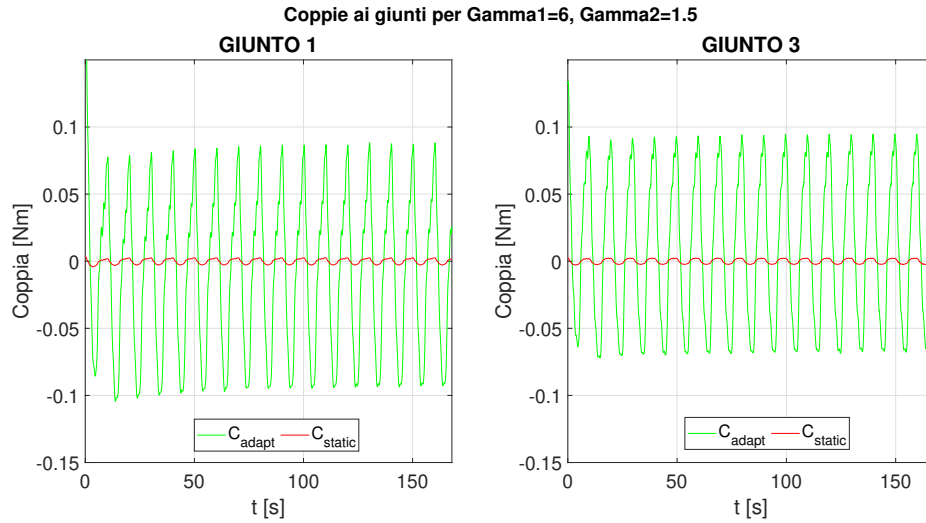


Figura 6.49: Contributi di coppia per $\Gamma_1=6$; $\Gamma_2= 1.5$

Anche inserendo il termine di *forgetting* è stato possibile garantire il rispetto delle condizioni imposte dal criterio di "taratura differenziata" sull'errore e sul rapporto tra contributi.

Considerazioni analoghe a quanto visto per i tre cavi valgono evidentemente

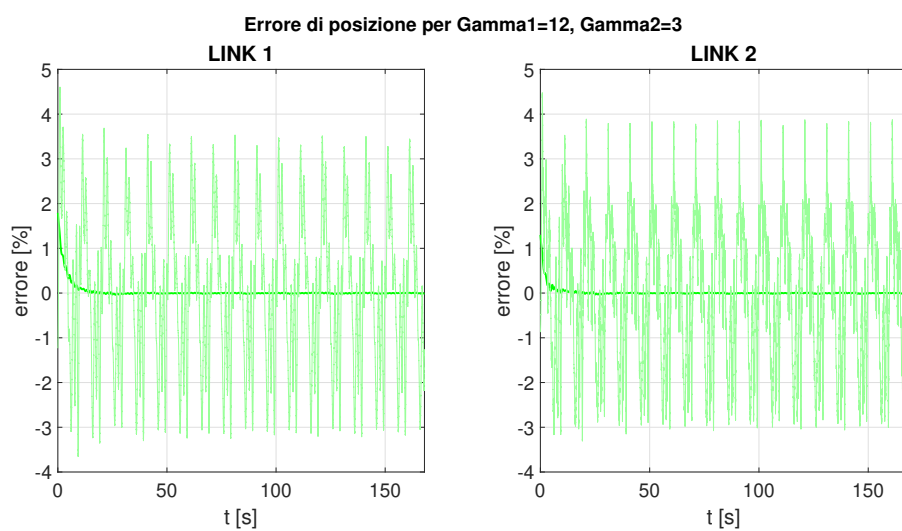


Figura 6.50: Errore di posizione per $\Gamma_1=12$; $\Gamma_2= 3$

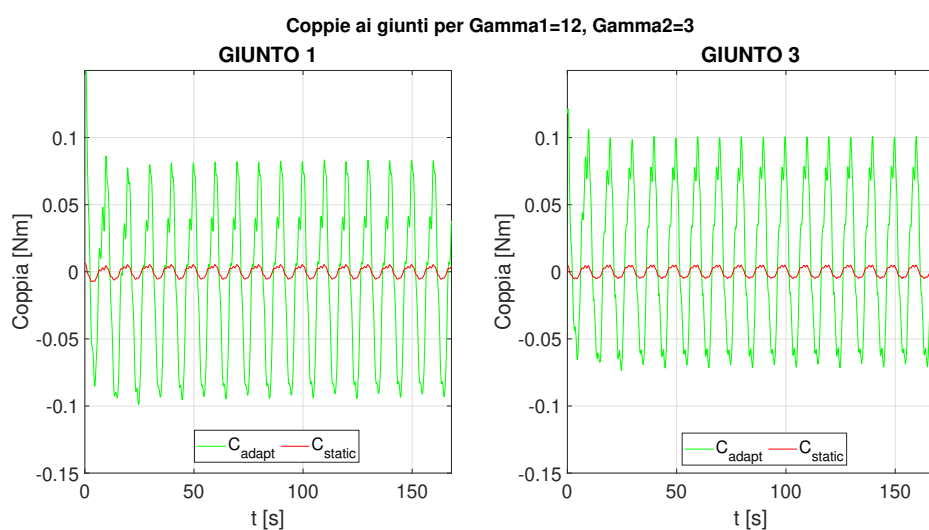
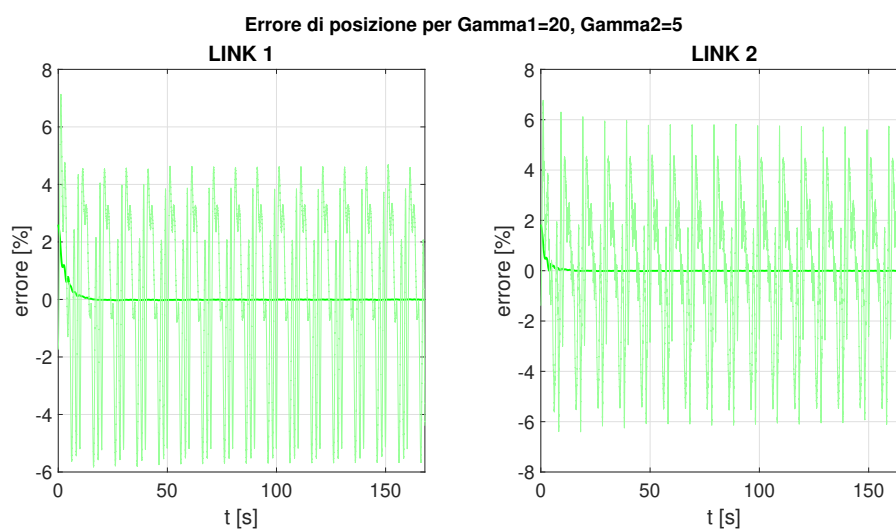
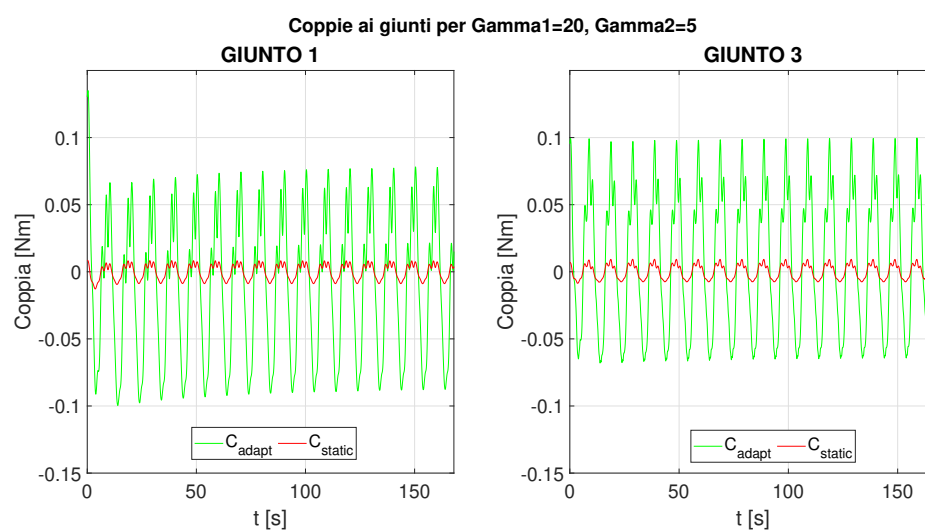


Figura 6.51: Contributi di coppia per $\Gamma_1=12$; $\Gamma_2= 3$

Figura 6.52: Errore di posizione per $\Gamma_1=20$; $\Gamma_2= 5$ Figura 6.53: Contributi di coppia per $\Gamma_1=20$; $\Gamma_2= 5$

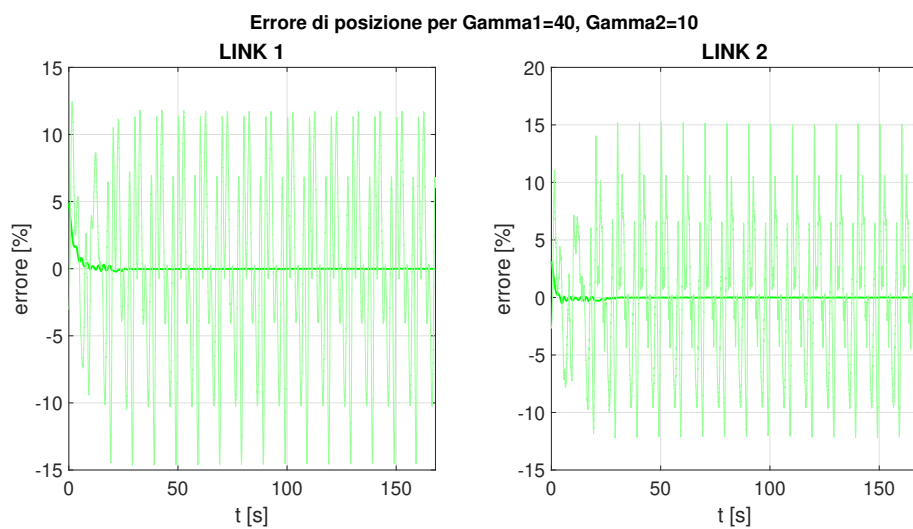


Figura 6.54: Errore di posizione per $\Gamma_1=40$; $\Gamma_2=10$

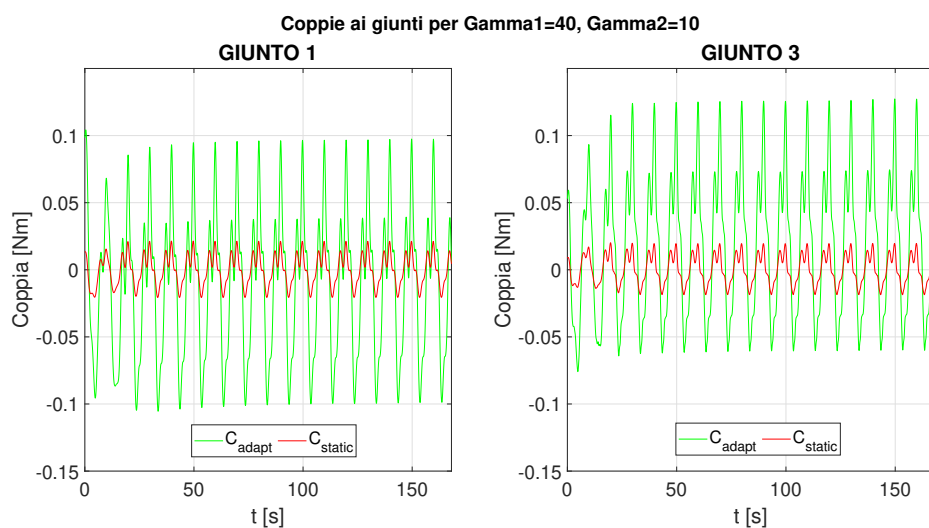
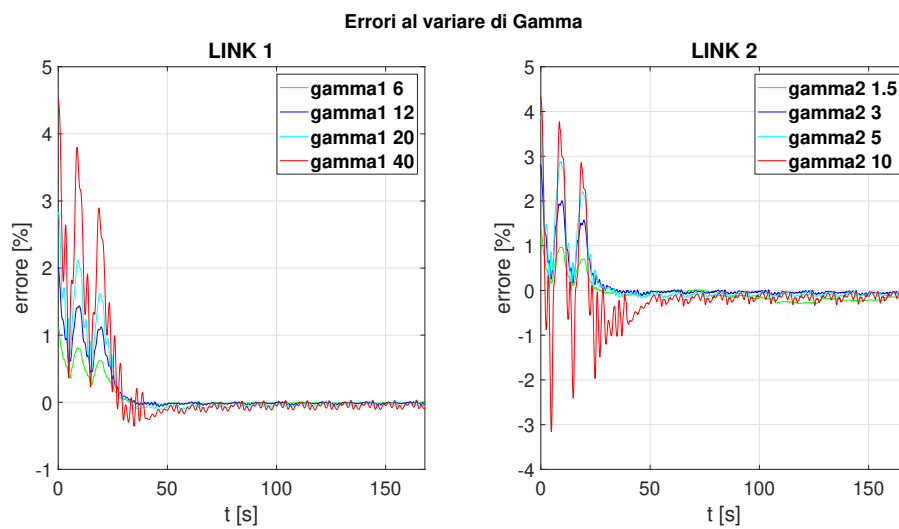
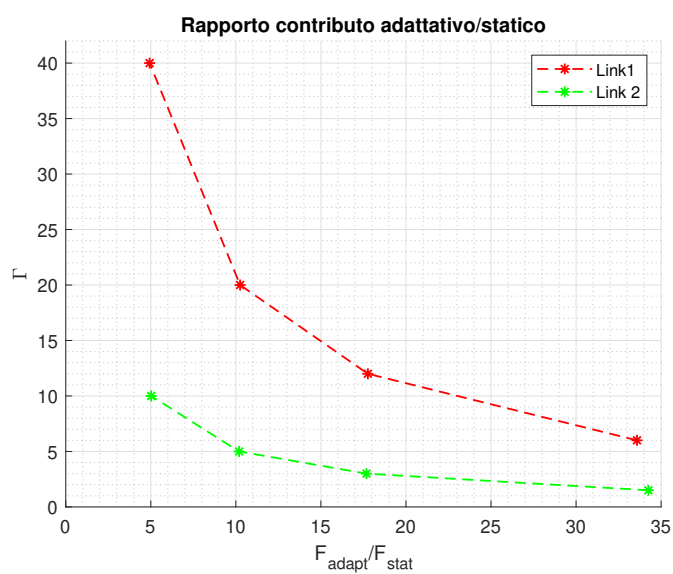


Figura 6.55: Contributi di coppia per $\Gamma_1=40$; $\Gamma_2=10$

Figura 6.56: Errore di posizione al variare di Γ Figura 6.57: Rapporto tra contributi C_{adapt}/C_{stat} al variare di Γ

anche per la parte di modellizzazione paziente, cui è stato necessario associare guadagni proporzionali distinti e con la stessa proporzione esistente tra i Kp_1 e Kp_2 del robot; per modellizzare un soggetto con discrete capacità di movimento sono stati adottati $Kp_{1pz}=0.03$ e $Kp_{2pz}=0.06$, impiegando il solito guadagno derivativo come un centesimo del proporzionale.

- **Comportamento complessivo robot e paziente**

I comportamenti osservati per i giunti in fase di prove sperimentali "complete", cioè effettuate impiegando il robot nella configurazione di adattativo con *forgetting* e di paziente simulato con capacità di apprendimento (in $n=12$ cicli), sono in linea con quelle già studiate per i cavi.

Si riportano dunque i grafici inerenti a un insieme di prove effettuate con due set di Γ e al variare di τ . Ciò che ci si aspetta è ormai noto: valori ridotti dell'*adaptive gain* rendono il controllore più veloce nella correzione dell'errore e, se abbinati a τ grandi, rendono il robot più "severo" e meno incline alla possibilità dell'errore sull'inseguimento determinata dall'azione del paziente, la quale risulterà limitata. Vale il viceversa, cioè un manipolatore più "accondiscendente" è innanzitutto costruito attorno a valori di costanti di tempo bassi, cioè ff elevati che fanno sì che questo dimentichi la dinamica appresa, quindi il comportamento del paziente; se ad esso si abbina poi un Γ alto allora si otterrà un manipolatore che lascia anche più "libertà temporale" all'azione del paziente, in quanto, anche se presente un errore nel tracking, non andrà a correggerlo troppo velocemente. Ancora una volta poi, sono confermati gli andamenti dell'errore iniziale e residuo al variare di Γ e mantenendo il medesimo ff e viceversa.

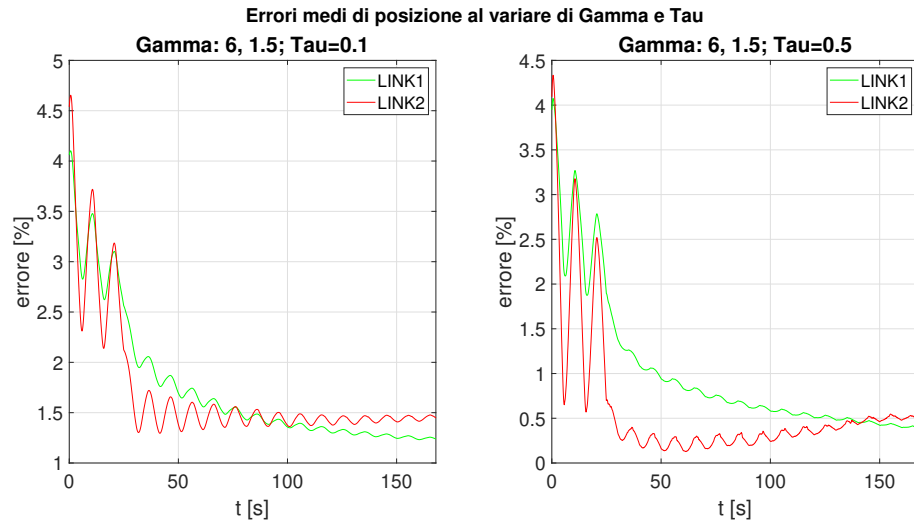


Figura 6.58: Errore di posizione per $\Gamma_1=6$ e $\Gamma_2=1.5$ al variare di τ

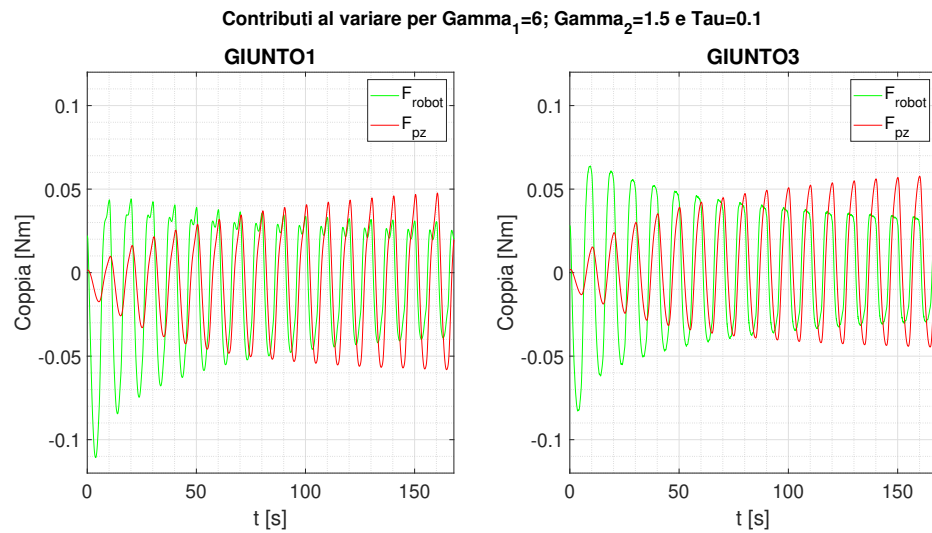


Figura 6.59: Contributi di coppia per $\Gamma_1=6$ e $\Gamma_2=1.5$ e $\tau=0.1$

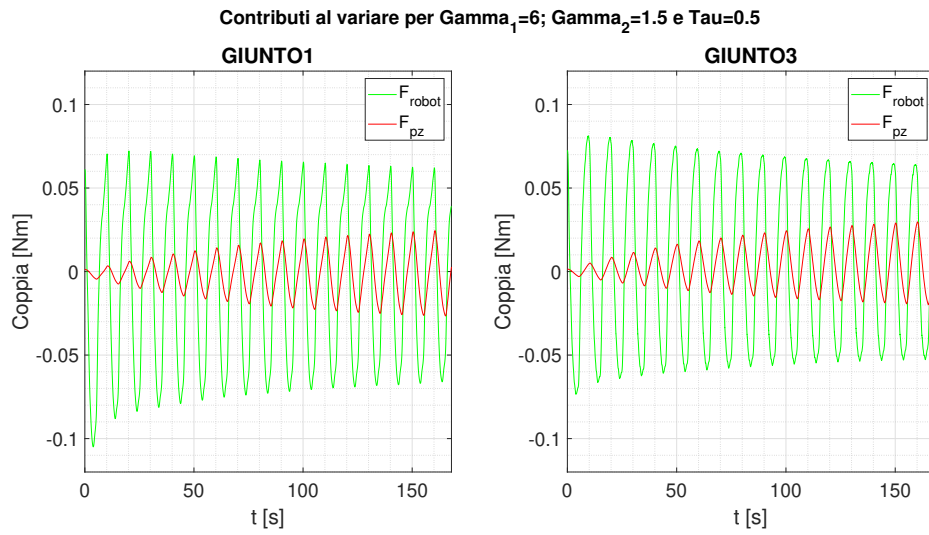


Figura 6.60: Contributi di coppia per $\Gamma_1=6$ e $\Gamma_2=1.5$ e $\tau=0.5$

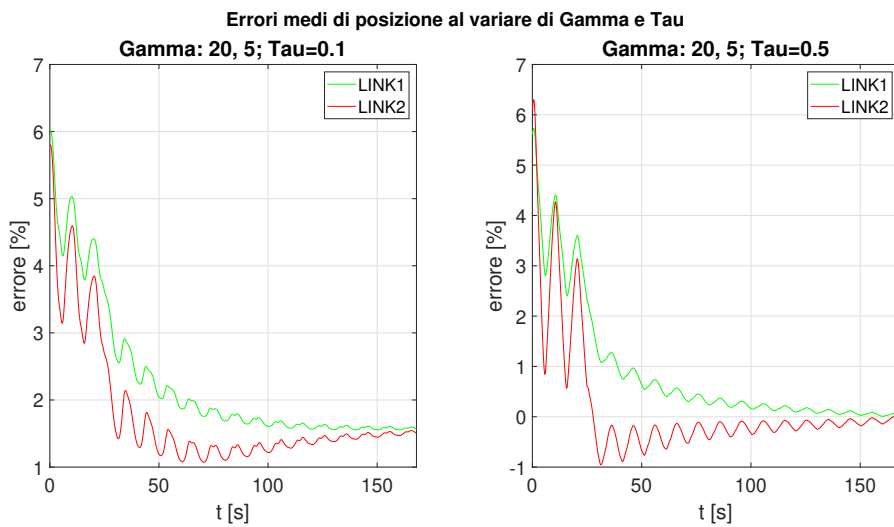


Figura 6.61: Errore di posizione per $\Gamma_1=20$ e $\Gamma_2=5$ al variare di τ

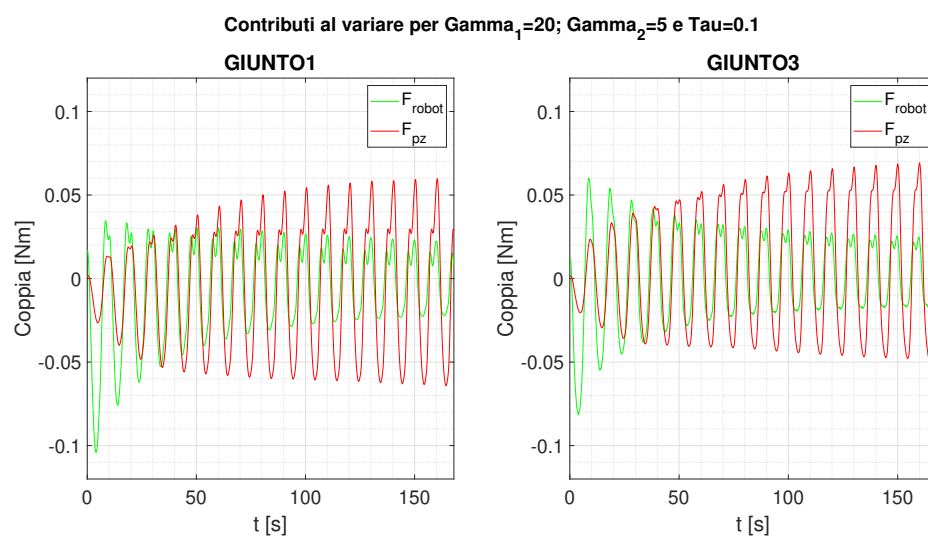


Figura 6.62: Contributi di coppia per $\Gamma_1=20$ e $\Gamma_2=5$ e $\tau=0.1$

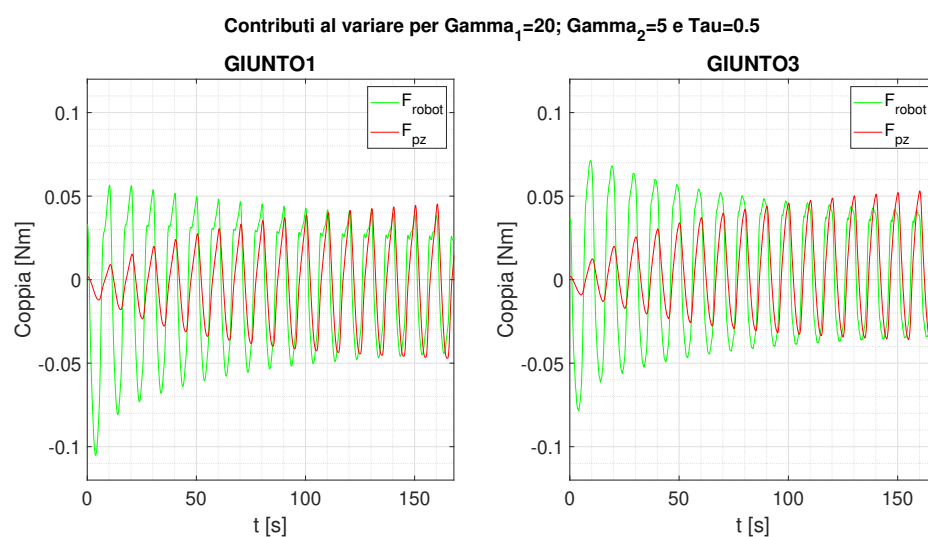


Figura 6.63: Contributi di coppia per $\Gamma_1=20$ e $\Gamma_2=5$ e $\tau=0.5$

Conclusioni

Questo lavoro di tesi si proponeva di esporre i passaggi che hanno permesso di migliorare il robot a cavi per la riabilitazione MariBot nella sua evoluzione MariBot 2.0.

A valle di una panoramica riguardante le differenze tra gli aspetti meccanici tra i due manipolatori, è stato presentato il nuovo schema elettrico e il relativo circuito di emergenza in bassa tensione. In questo contesto sono stati descritti i dispositivi e il nuovo cablaggio adottati, mirati alla realizzazione di un circuito più compatto e facente ausilio di elettronica moderna.

Un altro aspetto trattato in questo elaborato è stata la creazione della macchina a stati (*event manager*), necessaria per il controllo complessivo del manipolatore in tutte le fasi di esecuzione. Si è osservato come questo approccio, basato sull'individuazione di flag attivati al verificarsi di precise condizioni (dettate da opportune combinazioni di segnali di diversa natura) sia risultato efficace nel controllo della tensione di scrittura ai moduli di controllo dei diversi motori.

Il terzo tema trattato, fulcro dell'intero lavoro di tesi, è stato lo studio del controllo dei motori dei cavi e dei giunti, presentando le differenze tra un controllo a basso livello (PID), impiegato nella versione precedente del robot e un controllo con logica adattativa con *forgetting factor*, più adatto all'impiego in ambito medico in quanto consente una maggior partecipazione da parte del paziente cui è destinata la terapia.

Una volta eseguite prove simulate e sperimentali dei controllori a basso livello, col fine di verificarne il noto comportamento e che le coppie e forze misurate nella realtà fossero effettivamente in linea con quelle calcolate dalla teoria attraverso modelli dinamici, si è passati allo studio del controllo adattativo.

A valle di simulazioni svolte nel caso del singolo cavo per il movimento ciclico di una massa di prova, è stato possibile studiare il comportamento di questa particolari architetture di controllo e la variazione che i diversi parametri determinano nella forza sviluppata dal robot. Lo step successivo è stato provare sperimentalmente quanto testato in simulazione ed è stato possibile ottenere dei risultati del tutto confrontabili con quanto atteso, sia per quanto riguarda gli andamenti che gli ordini di grandezza.

Basandosi sui risultati ottenuti, sono state eseguite prove sperimentali per il controllo dei tre cavi a sostegno dell'ortesi e dei giunti, ottenendo in tutti i casi dei comportamenti in linea con quanto visto per il singolo cavo. Occorre sottolineare che in tutti i casi, per quantificare il contributo del paziente, è stato sempre considerato come simulato.

Nell'ottica di portare ulteriori migliorie e aggiornamenti a MariBot2.0, sotto l'aspetto del controllo, si potrebbe pensare di aggiungere una interfaccia che permetta l'inserimento dei parametri paziente e di terapia attraverso un'interfaccia più user friendly e comoda del metodo attuale, che consiste nel modificare i parametri direttamente dallo script *MATLAB*. Dal punto di vista dell'event manager, poi, il metodo della macchina a stati lascia molta libertà nell'aggiungere fasi di esecuzione: si potrebbe pensare, per esempio, di implementare la possibilità di mettere in pausa il ciclo di esercizio e successivamente farlo ripartire in base al comando del terapeuta, o ancora fare in modo che sia possibile cancellare uno dei punti acquisiti nel caso in cui lo stesso terapeuta lo desiderasse. Per quanto riguarda il controllo adattativo, infine, si può pensare di migliorarlo aggiungendo uno schema che consenta il mantenimento in tensione dei cavi, anche durante il ciclo di terapia. Nel caso di un paziente reale, infatti, non si può escludere che nel caso di movimento di alzata questo non sia capace di anticipare il movimento, determinando lo *slacking* dei cavi. Infine, ci si propone di sottoporre il robot a test clinici volti a validarne l'efficacia clinica tramite esercizi riabilitativi applicati a pazienti realmente affetti da ictus.

Appendice A

Implementazione pianificatore traiettoria

A.1 Implementazione pianificatore traiettoria

L'implementazione del pianificatore di traiettoria si può articolare in tre fasi:

- Ricerca delle posizioni da interpolare
- Precalcolo degli istanti di passaggio
- Generazione traiettoria di riferimento

Di seguito verranno riportati i codici per la realizzazione di ciascuna di queste parti.

A.1.1 Ricerca delle posizioni da interpolare

Le posizioni da usare come punti di partenza per la generazione di traiettoria si distinguono tra gli STARTER, ossia le ultime posizioni misurate alla pressione del pulsante che segna l'inizio della terapia e i punti di passaggio desiderati, acquisiti in fase di apprendimento. Gli STARTER vengono memorizzati come valore singolo mentre le posizioni acquisite sono salvate in un vettore di cinque elementi come descritto nel capitolo 4.

Questi, insieme agli intervalli di tempo definiti in 2), costituiscono le coordinate dei punti da interpolare per la generazione della traiettoria complessiva al punto 3).

A.1.2 Precalcolo degli istanti di passaggio

Per la natura intrinseca di *Simulink* risulta impossibile creare il riferimento pensando di creare un singolo ciclo e poi semplicemente replicarlo per un numero n_{cicli} di volte sfruttando un ciclo for. Questo approccio non viene correttamente eseguito in quanto l'esecuzione del programma *Real – Time* avviene con cadenza fissata corrispondente al tempo di campionamento (nel caso in esame 1 ms), il che si ripercuote nel calcolo della sola prima iterazione di un ciclo for.

Per risolvere questa problematica si è pensato di precalcolare gli istanti di cambio fase per ognuno degli n_{cicli} in modo da poter creare dei vettori "M" contenenti, per ogni parte del moto, il corretto istante temporale. Questi devono essere inizializzati a vettori nulli $[1, n_{cicli}]$ attraverso diversi blocchi di *memory*. Questi verranno riempiti attraverso un ciclo for che itera sulla variabile i e saranno successivamente letti dal pianificatore da un puntatore all'elemento j -esimo che vi assocerà un determinato valore di posizione (tra gli STARTER o tra le posizioni acquisite) in modo da ottenere i punti da usare per l'interpolazione.

```
function [cic_i_M, t_cic, t_setup_M, t_att_in_M, t_and_M, t_f_int_M, t_rit_M] = fcn(cic_i_M, n_cic, t_setup, t_att_in, t_fermo_int, t_corsa, t_setup_M, t_att_in_M, t_and_M, t_f_int_M, t_rit_M)
% calcola tempi ripetitivi di ogni fase della ciclica, lo fa contando su i
% fino a n_cicli (ho SUBITO la creazione dei tempi giusti: sarà un vettore
% di ampiezza variabile in base a n_cicli). Poi userò questi tempi
% calcolati come metodo di paragone per aggiornare una variabile j dentro
% al pianificatore

t_cic = t_att_in + t_fermo_int * 2 * t_corsa;

for i=1:n_cic
    % definizione numero del ciclo i-esimo eseguito
    ciclo_i=i;
    cic_i_M(i,:)=ciclo_i;
    % intervalli tempi di setup
    tf_setup = (t_setup) + (i-1)*t_cic;
    t_setup_M(i,:) = tf_setup;
    % intervalli tempi di attesa iniziale
    tf_att_iniz = (t_setup + t_att_in) + (i-1)*t_cic;
    t_att_in_M(i,:) = tf_att_iniz;
    % intervalli tempi di andata
    tf_andata = (t_setup + t_att_in + t_corsa) + (i-1)*t_cic;
    t_and_M(i,:) = tf_andata;
    % intervalli tempi di fermo intermedio
    tf_fermo_int = (t_setup + t_att_in + t_corsa + t_fermo_int) + (i-1)*t_cic;
    t_f_int_M(i,:) = tf_fermo_int;
    % intervalli tempi di ritorno
    tf_ritorno = (t_setup + t_att_in + t_corsa + t_fermo_int + t_corsa) + (i-1)*t_cic;
    t_rit_M(i,:) = tf_ritorno;
end
```

Figura A.1: Function di precalcolo degli istanti di tempo

A.1.3 Generazione traiettoria di riferimento

Il procedimento che verrà descritto è analogo per la creazione del riferimento per tutti i motori, a titolo di esempio si riporta solo il pianificatore del riferimento per LINK1.

Il pianificatore dispone ora di tutte le informazioni necessarie per la costruzione delle traiettorie secondo quanto descritto nel capitolo quattro (poli5, costante, spline cubica, costante, spline cubica, costante).

Occorre soffermarsi sull'implementazione della function *punti_{successivi}*, necessaria per garantire la corretta creazione della spline cubica che, secondo comando *MATLAB*, non può essere generata se i punti di interpolazione non sono diversi o crescenti: tale funzione distingue tutti i casi possibili e elimina il problema.

Lo script si basa sul continuo confronto tra il tempo di clock *clk_delayed* e i tempi memorizzati nei vettori *M* riferiti a un preciso ciclo *j*-esimo, inizializzato a 1. Oltre alla semplice traiettoria iniziale di setup e quella finale, occorre prestare maggior attenzione alla parte ciclica dell'esercizio. In base al risultato di tale confronto si è in grado di determinare in quale fase del ciclo in questione ci si trova e corrispondentemente si produrrà la traiettoria adeguata. Una volta trascorsa la durata del ciclo *j* si passa a quello successivo successivo, accedendo quindi al nuovo set di istanti temporali memorizzati in *M* e si procede in maniera analoga.

```

1 function [j,ref] = fcn( j,cic_i_M,n_cic, clk_rit, n_pti,pos_STARTER, pos_vec_acq, t_cic, t_setup_M, t_att_in_M, t_and_M, t_corsa, t_fermo_int_M, t_rit_M)
2
3     t = clk_rit;
4
5
6     % distinzione casi in base a punti di acquisizione successivi: se uguali spline non
7     % viene generata correttamente... bisogna evitare che si verifichi
8     [caso_ref, wp_ref] = punti_successivi(pos_vec_acq, n_pti);
9
10
11 %%% FASE 1: SETUP
12 if (t >= 0 && t < t_setup_M(1))
13     qi = pos_STARTER;
14     qf = pos_vec_acq(1,1);
15     T_AND = t_setup_M(1);
16     n_type = 5;
17     ref = traj_ppMOD(qi , qf , T_AND , n_type, t);
18
19 %%% FASE 2: CICLICA
20 elseif t >= t_setup_M(1) && t < t_setup_M(1)+n_cic*t_cic
21
22
23     % FASE 2.1: ATTESA_INIZIALE
24     if t >= t_setup_M(cic_i_M(j)) && t < t_att_in_M(cic_i_M(j))
25         ref=pos_vec_acq(1,1);
26
27     % FASE 3.2: ANDATA
28     elseif t >= t_att_in_M(cic_i_M(j)) && t < t_and_M(cic_i_M(j))
29         t_sp_AND = t-t_att_in_M(cic_i_M(j)); % spline andata dovrà essere valutata su un clk ulteriormente ritardato che inizia al termine della fase di attesa iniziale di ogni ciclo
30         T_AND = t_corsa;
31
32         % creazione traj di andata
33         if caso_ref == 1
34             ref = pos_vec_acq(1,1);
35
36         elseif caso_ref == 2
37             way_points_AND = wp_ref;
38             t1_AND = 0;
39             t2_AND = T_AND;
40             time_points_AND = [t1_AND, t2_AND];
41             cs_andata = spline(time_points_AND, [0 way_points_AND 0]);
42             t_andata = t_sp_AND;
43             ref = ppval(cs_andata, t_andata);
44
45         elseif caso_ref == 3
46             way_points_AND = wp_ref;
47             % calcolo delta di posizione tra punti di acquisizione successivi e delta totale
48             delta1_AND = abs((way_points_AND(2))-way_points_AND(1));
49             delta2_AND = abs((way_points_AND(n_pti))-way_points_AND(2));
50             deltaTOT_AND = delta1_AND+delta2_AND;
51             % costruzione vettore dei tempi
52             t1_AND = 0;
53             t3_AND = T_AND;
54             t2_AND = delta1_AND*t3_AND/deltaTOT_AND;
55             time_points_AND = [t1_AND, t2_AND, t3_AND];
56             cs_andata = spline(time_points_AND, [0 way_points_AND 0]);
57             t_andata = t_sp_AND;
58             ref = ppval(cs_andata, t_andata);
59
60         elseif caso_ref == 4
61             way_points_AND = wp_ref;
62             % calcolo delta di posizione tra punti di acquisizione successivi e delta totale
63             delta1_AND = abs((way_points_AND(2))-abs(way_points_AND(1)));
64             delta2_AND = abs((way_points_AND(3))-abs(way_points_AND(2)));
65             delta3_AND = abs((way_points_AND(n_pti))-abs(way_points_AND(3)));
66             deltaTOT_AND = delta1_AND+delta2_AND+delta3_AND;
67             % costruzione vettore dei tempi
68             t1_AND = 0;
69             t4_AND = T_AND;
70             t2_AND = delta1_AND*t4_AND/deltaTOT_AND;
71             t3_AND = delta2_AND*t4_AND/deltaTOT_AND + t2_AND;
72             time_points_AND = [t1_AND, t2_AND, t3_AND, t4_AND];
73             cs_andata = spline(time_points_AND, [0 way_points_AND 0]);
74             t_andata = t_sp_AND;
75             ref = ppval(cs_andata, t_andata);
76
77         elseif caso_ref == 5
78             way_points_AND = wp_ref;
79             % calcolo delta di posizione tra punti di acquisizione successivi e delta totale
80             delta1_AND = abs((way_points_AND(2))-way_points_AND(1));
81             delta2_AND = abs((way_points_AND(3))-way_points_AND(2));
82             delta3_AND = abs((way_points_AND(4))-way_points_AND(3));
83             delta4_AND = abs((way_points_AND(n_pti))-way_points_AND(4));
84             deltaTOT_AND = delta1_AND+delta2_AND+delta3_AND+delta4_AND;
85             % costruzione vettore dei tempi
86             t1_AND = 0;
87             t5_AND = T_AND;
88             t2_AND = delta1_AND*t5_AND/deltaTOT_AND;
89             t3_AND = delta2_AND*t5_AND/deltaTOT_AND + t2_AND;
90             t4_AND = delta3_AND*t5_AND/deltaTOT_AND + t3_AND;
91             time_points_AND = [t1_AND, t2_AND, t3_AND, t4_AND, t5_AND];
92             cs_andata = spline(time_points_AND, [0 way_points_AND 0]);
93             t_andata = t_sp_AND;
94             ref = ppval(cs_andata, t_andata);
95         else
96             ref = 0;
97     end

```

Figura A.2: Function per pianificatore, parte 1

```

100
101
102 % FASE 2.3: FERMO_INTERMEDIO
103 elseif t >= t_fermo_int_M(cic_i_M(j)) && t < t_fermo_int_M(cic_i_M(j+1))
104     ref=pos_vec_acq(1,n_pti);
105
106
107
108 % FASE 2.4: RITORNO
109 elseif t >= t_fermo_int_M(cic_i_M(j)) && t < t_rit_M(cic_i_M(j))
110     t_sp_RIT = t-t_fermo_int_M(cic_i_M(j)); % spline ritorno dovrà essere valutata su un cik ulteriormente ritardato
111                                           % che inizia al termine della fase di fermo intermedio di ogni ciclo
112     T_RIT = t_corsa;
113
114 % creazione traj di ritorno
115 if caso_ref == 1
116     ref = pos_vec_acq(1,1);
117
118 elseif caso_ref == 2
119     way_points_RIT = flip(wp_ref);
120     t1_RIT = 0;
121     t2_RIT = T_RIT;
122     time_points_RIT = [t1_RIT, t2_RIT];
123     cs_ritorno = spline(time_points_RIT, [0 way_points_RIT 0]);
124     t_ritorno = t_sp_RIT;
125     ref = ppval(cs_ritorno, t_ritorno);
126
127 elseif caso_ref == 3
128     way_points_RIT = flip(wp_ref);
129     % calcolo delta di posizione tra punti di acquisizione successivi e delta totale
130     delta1_RIT = abs((way_points_RIT(2))-(way_points_RIT(1)));
131     delta2_RIT = abs((way_points_RIT(3))-(way_points_RIT(2)));
132     deltaTOT_RIT = delta1_RIT+delta2_RIT;
133     % costruzione vettore dei tempi
134     t1_RIT = 0;
135     t2_RIT = T_RIT;
136     t3_RIT = delta1_RIT*t2_RIT/deltaTOT_RIT;
137     time_points_RIT = [t1_RIT, t2_RIT, t3_RIT];
138     cs_ritorno = spline(time_points_RIT, [0 way_points_RIT 0]);
139     t_ritorno = t_sp_RIT;
140     ref = ppval(cs_ritorno, t_ritorno);
141
142 elseif caso_ref == 4
143     way_points_RIT = flip(wp_ref);
144     % calcolo delta di posizione tra punti di acquisizione successivi e delta totale
145     delta1_RIT = abs((way_points_RIT(2))-(way_points_RIT(1)));
146     delta2_RIT = abs((way_points_RIT(3))-(way_points_RIT(2)));
147     delta3_RIT = abs((way_points_RIT(4))-(way_points_RIT(3)));
148     deltaTOT_RIT = delta1_RIT+delta2_RIT+delta3_RIT;
149     % costruzione vettore dei tempi
150     t1_RIT = 0;
151     t4_RIT = T_RIT;
152     t2_RIT = delta1_RIT*t4_RIT/deltaTOT_RIT;
153     t3_RIT = delta2_RIT*t4_RIT/deltaTOT_RIT + t2_RIT;
154     time_points_RIT = [t1_RIT, t2_RIT, t3_RIT, t4_RIT];
155     cs_ritorno = spline(time_points_RIT, [0 way_points_RIT 0]);
156     t_ritorno = t_sp_RIT;
157     ref = ppval(cs_ritorno, t_ritorno);
158
159 elseif caso_ref == 5
160     way_points_RIT = flip(wp_ref);
161     % calcolo delta di posizione tra punti di acquisizione successivi e delta totale
162     delta1_RIT = abs((way_points_RIT(2))-(way_points_RIT(1)));
163     delta2_RIT = abs((way_points_RIT(3))-(way_points_RIT(2)));
164     delta3_RIT = abs((way_points_RIT(4))-(way_points_RIT(3)));
165     delta4_RIT = abs((way_points_RIT(5))-(way_points_RIT(4)));
166     deltaTOT_RIT = delta1_RIT+delta2_RIT+delta3_RIT+delta4_RIT;
167     % costruzione vettore dei tempi
168     t1_RIT = 0;
169     t5_RIT = T_RIT;
170     t2_RIT = delta1_RIT*t5_RIT/deltaTOT_RIT;
171     t3_RIT = delta2_RIT*t5_RIT/deltaTOT_RIT + t2_RIT;
172     t4_RIT = delta3_RIT*t5_RIT/deltaTOT_RIT + t3_RIT;
173     time_points_RIT = [t1_RIT, t2_RIT, t3_RIT, t4_RIT, t5_RIT];
174     cs_ritorno = spline(time_points_RIT, [0 way_points_RIT 0]);
175     t_ritorno = t_sp_RIT;
176     ref = ppval(cs_ritorno, t_ritorno);
177
178 else
179     ref = 0;
180 end
181
182
183 % AGGIORNAMENTO PUNTIACQUISIZIONE j ALLA MATRICE ciclo_i_M
184 elseif round(t)==round(t_rit_M(cic_i_M(j)))
185     ref=pos_vec_acq(1,1); %comunque quella finale del ritorno 2.4)
186     j=j+1;
187 else
188     ref = 0;
189 end
190
191
192 % FASE 3: ATTESA_FINALE
193 ref=pos_vec_acq(1,1);
194
195
196
197 end
198
199 end

```

Figura A.3: Function per pianificatore, parte 2

Bibliografia

- [1] S. I. di Neurologia, “Ictus,” <https://neuro.it/web/eventi/NEURO/index.cfm>, Online, 2021.
- [2] V. o. AULSS 4, “Cura dell’ictus - i percorsi riabilitativi dell’ulss,” <https://www.aulss4.veneto.it/>, Online, 2012.
- [3] G.U., “Linee guida del ministro della sanità per le attività di riabilitazione,” <https://www.aulss4.veneto.it/>, Gazzetta Ufficiale, 1998.
- [4] M. L. Lo, Kenneth; Stephenson and Craig, “Effectiveness of robotic assisted rehabilitation for mobility and functional ability in adult stroke patients: a systematic review,” *JBIC Database of Systematic Reviews and Implementation Reports*, vol. 15(12), pp. 3049–3091, 2017.
- [5] K. Kwakkel and Krebs, “Effects of robot-assisted therapy on upper limb recovery after stroke: a systematic review,” *Neurorehabilitation and neural repair*, vol. 22, p. 111–121, 2008.
- [6] G. Bonaiuti, Calabrò, “Il mercato della robotica per la riabilitazione: stato dell’arte e prospettive,” *Giornale italiano di Medicina Riabilitativa*, vol. 36(3), pp. 79–81, 2021.
- [7] O. delle Nazioni Unite, “Global issues population,” <https://www.un.org/en/global-issues/population>, Online, 2022.
- [8] Xiong and Diao, “A review of cable-driven rehabilitation devices,” *Disability and Rehabilitation: Assistive Technology*, vol. 15(8), pp. 885–887, 2020.

- [9] R. Vitiello, Lenzi *et al.*, “Neuroexos: A powered elbow exoskeleton for physical rehabilitation,” *IEEE Transactions on Robotics*, vol. 29, p. 220–235, 2013.
- [10] S. Banala, Agrawal, “Active leg exoskeleton (alex) for gait rehabilitation of motor-impaired patients,” in *2007 IEEE 10th International Conference on Rehabilitation Robotics*, vol. -, pp. 401–407, 2007.
- [11] A. Krebs, Hogan and Volpe, “Robot-aided neurorehabilitation,” *IEEE transactions on rehabilitation engineering : a publication of the ieee engineering in medicine and biology society*, vol. 6(1), p. 75–87, 1998.
- [12] P. Hidler, Nichols and Brady, “Robot-aided neurorehabilitation,” *Advances in the understanding and treatment of stroke impairment using robotic devices*, vol. 12, pp. 22–35, 2005.
- [13] M. Rosati, Gallina and Rossi, “Design of a new 5 d.o.f. wire-based robot for rehabilitation,” in *9th International Conference on Rehabilitation Robotics*, vol. -, p. 430–433, 2005.
- [14] “ser. asme international mechanical engineering congress and exposition,” *vol. Volume 2: Biomedical and Biotechnology Engineering*, vol. -, pp. –, 2008.
- [15] R. Masiero, Celia and Armani, “Robotic-assisted rehabilitation of the upper limb after acute stroke,” *Archives of physical medicine and rehabilitation*, vol. 88, p. 142–149, 2007.
- [16] R. e. a. Masiero, M. Armani, “Upper-limb robot-assisted therapy in rehabilitation of acute stroke patients: focused review and results of new randomized controlled trial,” *J Rehabil Res Dev*, vol. 48, p. 355–366, 2011.
- [17] Hire, “Arjo marisa. disability hoist hire,” <https://www.disabilityhoisthire.co.uk/>, Online, 2021.
- [18] Tessarolo, “Progetto di controllo di robot con bus di campo,” pp. 82-97, Università degli studi di Padova, 2005-2006.

-
- [19] Rossi, “Ottimizzazione cinematica di un sistema robotico per la riabilitazione,” pp. 110-115, Università degli studi di Padova, 2021-2022.
- [20] R. Hogan, Krebs *et al.*, “Motions or muscles? some behavioral factors underlying robotic assistance of motor recovery,” *J. Rehabil. Res. Dev.*, vol. 43, p. 605, 2006.
- [21] C. Jezernik, Scharer and Morari¹, “Adaptive robotic rehabilitation of locomotion: a clinical study in spinally injured individuals,” *Spinal Cord*, vol. 41, p. 657–666, 2003.
- [22] R. Wolbrecht, Chan and Bobrow, “Optimizing compliant, model-based robotic assistance to promote neurorehabilitation,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 16, pp. 286–297, 2008.

Ringraziamenti

Questa tesi, per me, significa tanto.

Ok, simboleggia la fine del mio percorso di studi.

Percorso lungo, difficile, pieno di sacrifici. Pieno di momenti in cui ho pensato (come tutti gli aspiranti ingegneri, temo) di mollare. Di lasciar perdere. Sono state le piccole grandi soddisfazioni raccolte qua e là durante questo cammino e l'ENORME supporto che mi hanno SEMPRE dato TUTTE le persone che mi vogliono e hanno voluto bene che mi ha permesso di arrivare a scrivere queste ultime righe.

Ma non è solo questo.

Per me questa tesi è la "conferma".

In tutti questi mesi passati in laboratorio su codici, schede, motori, sbattendo la testa per trovare soluzioni per "far muovere qualcosa da solo", ho finalmente capito che QUESTA è DAVVERO la mia strada.

Vorrei allora ringraziare il Professor Boschetti, innanzitutto per la avermi dato la possibilità di svolgere questo lavoro e in secondo luogo per la disponibilità sempre dimostrata.

Ringrazio poi di cuore Giacomo Zuccon. Il tuo aiuto, i tuoi consigli, le tue correzioni, la tua pazienza, la tua enorme disponibilità (non so quante persone oltre a te avrebbero voluto seguire un tesista da oltreoceano) sono stati fondamentali per me per ottenere il massimo possibile da questa tesi. Ti ringrazio anche per la simpatia e l'affetto con cui mi hai sempre trattato: hanno reso questi 6 mesi di studio decisamente più vivibili, tra un "buildo" e un "ti-ribuildo".

Non posso non ringraziare poi tutto il gruppo di altri Dottorandi e Ricercatori del laboratorio di robotica di Padova. Il vostro contributo è spesso stato illuminante in alcuni punti del mio lavoro: Riccardo Minto (ti penserò ogni volta che dovrò usare un blocco "Memory" su Simulink), Michele Tonan (sei finalmente libero di riprenderti il driver che mi hai prestato, sono stato bravo e non ho bruciato anche questo, hai visto?), Domenico Donà (grazie per aver chiarito punti a me oscuri della dinamica inversa), Matteo Bottin (spesso

quattro chiacchiere con te mi hanno aiutato nel risolvere dubbi amletici).

Poi c'è la mia famiglia.

Mamma. Papà. Giulia. Nonna.

Sapete bene come ho passato questi anni di università. Sapete bene che sfida sia stata per me. E per questo avete sempre fatto di tutto per darmi una mano, per supportarmi (anche quando, mi rendo conto, potevo risultare molto poco meritevole di supporto).

La famiglia non si sceglie.

E io sarò sempre grato al destino per avermi dato voi. Non potrei desiderare una famiglia migliore di questa.

(Sto parlando anche a voi, lassù).

Poi ci sei tu, Altea.

Abbiamo condiviso tutto, dalle superiori fino a qui (persino la data della discussione. Decisamente scomodo, ma poteva finire diversamente, se ci pensi?).

Siamo cresciuti insieme, abbiamo affrontato le stesse paure, gli stessi scogli, anno dopo anno.

Anche per te, non so come ringraziarti per tutto quello che hai sempre fatto per aiutarmi e per essermi sempre stata vicino.

Per avermi sempre aspettato.

Per avermi sempre capito.

Per aver passato le ultime serate in silenzio insieme a me, standomi solo accanto finché mi disperavo per dare un senso ai mille grafici di questa tesi.

La propria compagna, è vero, si può scegliere.

E io risceglierei te.