ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA

FACOLTÀ DI INGEGNERIA E ARCHITETTURA

DIPARTIMENTO

Ingegneria Industriale - DIN

CORSO DI LAUREA MAGISTRALE

Ingegneria Meccanica con Curriculum Biomeccanica

Simulazioni predittive del cammino mediante modelli di dinamica muscoloscheletrica

CANDIDATO

Paolo Riccioni

RELATORE:

Prof. Marco Viceconti

CORRELATORE:

Dott. Giorgio Davico

Anno Accademico 2021/2022

Sessione III

Sommario

Abstract

Le simulazioni computazionali di dinamica del movimento mediante modelli muscoloscheletrici consentono di stimare quantità non misurabili direttamente e di valutare/studiare scenari clinici e/o chirurgici ipotetici. Quest'ultimo aspetto è trattato con simulazioni basate sul Forward Approach, tramite cui è possibile stimare un movimento (cinematica) a partire da un comando neurale noto. Il sistema nervoso centrale risolve un problema complesso e composto di vari sotto obiettivi, tecniche basate sulla teoria dei controlli e metodi di Optimal Control sono impiegate per risolvere il problema del controllo motorio mediante simulazioni predittive. Il software OpenSim Moco, mediante la collocazione diretta, permette l'utilizzo di metodi avanzati di optimal control per descrivere il controllo motorio tramite una funzione costo composta da più *sotto task*, da minimizzare rispettando vincoli cinematici e dinamici del sistema multicorpo.

L'obiettivo dell'elaborato è la realizzazione di una simulazione predittiva di camminata, con modelli muscoloscheletrici. È stato realizzato un workflow caratterizzato da due step: realizzazione di una condizione iniziale e di una simulazione predittiva. Il primo step è ottenuto tracciando file sperimentali, per generare cinematica e attivazioni del movimento più realistiche. Nel secondo step è realizzato un problema di optimal control con l'obiettivo di ottenere una camminata 'normale' aggiungendo progressivamente *Goals* da minimizzare. Lo studio esplorativo ha permesso di valutare il software, risultato intuitivo, - grazie all'interfaccia "user friendly" - e personalizzabile, e di ottenere cinematiche 'verosimili' con attivazioni in miglioramento progressivo all'aumentare della complessità della funzione costo. Tuttavia, alcune problematiche sono emerse (es. attivazioni differenti dall'EMG e ad angoli di flesso-estensione di caviglia molto differenti dalla cinematica di confronto), che saranno oggetto di valutazioni future.

Capitolo 1: Optimal Control

I grandi filosofi greci del passato non conoscevano l'espressione "controllo motorio"; erano interessati, tuttavia, alle origini dei movimenti biologici mirati e si concentrarono sulla relazione tra il corpo (che si muove) e l'anima (che controlla). Le seguenti due domande sono state comunemente discusse: Qual è l'origine del movimento nel mondo? Cosa spinge l'anima a comandare al corpo di muoversi?

Pitagora (571? – 497? a.C.) vedeva il movimento come evoluzione di numeri metafisici. L'anima è stata definita come un numero con la capacità di muoversi da solo.
Eraclezio di Efeso (540 - ??? a.C.) fondò una scuola nota come "quelli che credono nel movimento". Secondo i filosofi di quella scuola, gli oggetti erano al di là della conoscenza umana, solo le loro traiettorie potevano essere osservate.
Democrito (460 - 370? a.C.) giunse alla conclusione che il movimento dell'anima era trasmesso al corpo dal movimento degli atomi.
Secondo Platone (428 - 347 a.C.), l'auto-movimento era un segno di un'anima immortale, che avrebbe dovuto possedere una capacità unica di essere mossa da sola (simile a Pitagora). Platone paragonava i movimenti volontari umani a un carro mosso da cavalli controllati dall'anima.
Aristotele (384 - 322 a.C.) sosteneva che per ogni movimento dovevano esserci un motore e un corpo in movimento. Solo i corpi in movimento sono osservabili, mentre il motore (l'anima) no. [1]

1.1 Introduzione al controllo motorio e alla simulazione muscoloscheletrica computazionale

Il controllo motorio può essere definito come la generazione e l'attuazione di qualsiasi tipologia di movimento (muoversi, vedere, parlare, respirare, pensare, ecc.). A coordinare tutte le funzioni degli organi è preposto il sistema nervoso centrale (SNC).

Il SNC è composto da neuroni il cui scopo è raccogliere, elaborare e inviare informazioni agli "attuatori" impartendo ordini. Il segnale nervoso è un segnale di tipo chimico che presenta una velocità di conduzione lenta (0.5 – 2.0 m/s) rispetto al necessario per compiere atti motori repentini (es. saltare). È pertanto presente un secondo meccanismo di azione, anticipatorio e basato su apprendimento pregresso, per garantire maggior velocità e compiere azioni veloci: il sistema dei riflessi.
Il controllo motorio si divide quindi in controllo volontario, generato dalla coordinazione motoria volontaria e dalla propriocezione, e controllo involontario, generato dall' arco riflesso[1] o dalla risposta a uno stimolo percettivo.

Nel corso del XIX e XX secolo, il meccanismo di controllo motorio è stato al centro del dibattito scientifico. Numerose teorie sono state enunciate, a partire da quella proposta da Ivan Pavlov (1849-1936) sull'esistenza dei riflessi acquisiti (Conditioned Reflex [2]), secondo la quale il movimento è generato da una combinazione di riflessi. Charles Sherrington (1857-1952) suggerì invece che il movimento fosse prodotto dalla modulazione volontaria dei riflessi, mentre Graham Brown (1882-1965) aprì una linea di ricerca che, ignorando i riflessi, portò alla definizione della teoria del cosiddetto "Central pattern generator" (cioè i pattern essenziali del passo possono essere generati nel midollo spinale senza alcun segnale discendente dalla corteccia motoria). Nel XX secolo, Nikolai Bernstein (1896-1966) sviluppò infine la teoria degli engrammi, "programmi motori" archiviati in memoria, basato sullo studio di lavoratori che compiono pattern motori ripetitivi ad alta specializzazione che possono essere memorizzati in strutture anatomiche.

[1] Arco riflesso: i nevi afferenti ed efferenti di gruppi di muscoli si connettono nel midollo spinale che, oltre a scambiare informazioni con il sistema nervoso centrale generano anelli di retroazione locali responsabili dei riflessi.

In generale, quindi, le teorie del controllo motorio possono essere suddivise in teorie che si basano sul controllo di forze e/o momenti, di attivazioni e sulla teoria dei controlli.
Il controllo delle forze e/o momenti muscolari presuppone che il sistema nervoso centrale esegua calcoli che riflettono le interazioni tra il corpo e l'ambiente, e tra le parti del corpo stesso [3]. Il controllo con attivazioni muscolari presuppone invece che il sistema nervoso centrale possa elaborare e mandare patterns di input presinaptici a specifici gruppi di alfa-motoneuroni[2]. La meccanica del movimento risulta quindi influenzata, oltre che da questi input, anche dalle caratteristiche input-output di gruppi di motoneuroni, le proprietà del muscolo e le interazioni tra le parti del corpo in movimento e tra corpo e ambiente [4].
Il controllo con approccio alla teoria dei controlli ha come presupposto che il sistema per la produzione del movimento possa essere suddiviso in un controllore e una parte controllata e che il controllore neurale esegua certe operazioni computazionali per permettere il movimento [5].

Con l'apparizione dei primi modelli computerizzati (oggi chiamati digital twins) del sistema muscoloscheletrico [6], [7] e con il conseguente loro utilizzo per simulazioni di natura biomeccanica, il ruolo computazionale nell'applicazione delle teorie sul controllo motorio ha preso sempre più spazio, diventando per certi versi fondamentale.
Simulazioni computazionali di dinamica del movimento completano sempre più spesso gli esperimenti *in vivo*, consentendo di stimare quantità che non possono essere misurate direttamente (come le forze muscolari responsabili di un movimento osservato, i carichi articolari, le deformazioni dei tendini ecc.) e di valutare/studiare scenari clinici e/o chirurgici ipotetici (es. per valutare gli effetti di un intervento chirurgico) [8]. A tal fine, un flusso di lavoro tipico prevede di partire da una serie di dati sperimentali raccolti sul soggetto d'interesse, ad esempio all'interno di un laboratorio di analisi del movimento. Una volta elaborati, i dati vengono utilizzati per informare o guidare le simulazioni. In particolare, viene prima risolta la cinematica del sistema (modello muscoloscheletrico), e, a seguire, attraverso la risoluzione delle equazioni della dinamica vengono stimati i momenti articolari. Tramite l'ottimizzazione statica è possibile stimare le forze muscolari risolvendo il problema di ridondanza muscolare (i.e. più forze muscolari sconosciute rispetto ai vincoli del momento articolare). Un approccio di questo tipo è definito

[2] Alfa-motoneuroni: cellula nervosa di grande diametro che trasmette alla periferia gli impulsi motori innervando un numero variabile di fibre muscolari striate.

approccio inverso (Inverse Approach) in quanto le forze dei muscoli tali da generare il movimento osservato sono determinate a partire dalle misurazioni di movimento. In alternativa, in assenza di dati sperimentali, imponendo un set di eccitazioni/attivazioni muscolari (controllo neurale), da cui si ricavano le forze muscolari, è possibile stimare il movimento del modello. Questo secondo approccio (Forward Approach), meno vincolato a dati sperimentali, risulta essere utile per testare scenari ipotetici quali gli effetti di un intervento protesico o di un programma di rinforzo muscolare.

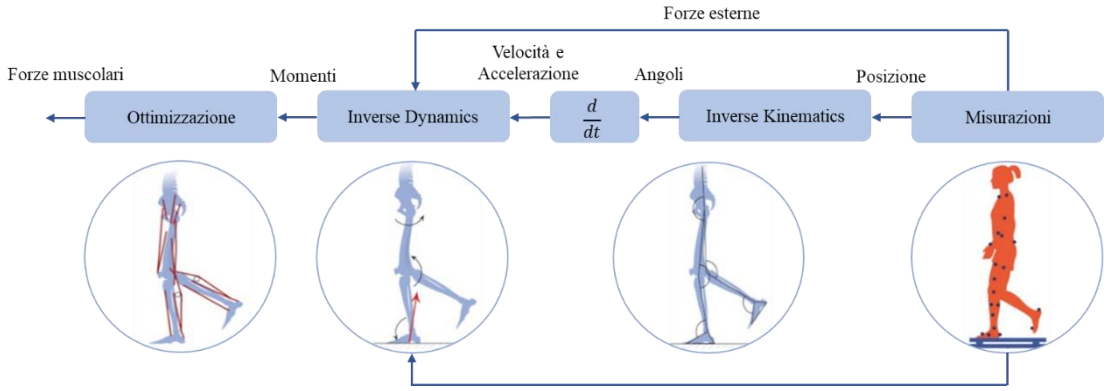

Figura 1: Inverse Approach. Rappresenta il flusso di lavoro per realizzare una simulazione muscoloscheletrica partendo da misurazioni sperimentali di movimento, per arrivare alla stima delle forze muscolari passando attraverso analisi di Inverse Kinematics e Inverse Dynamics.

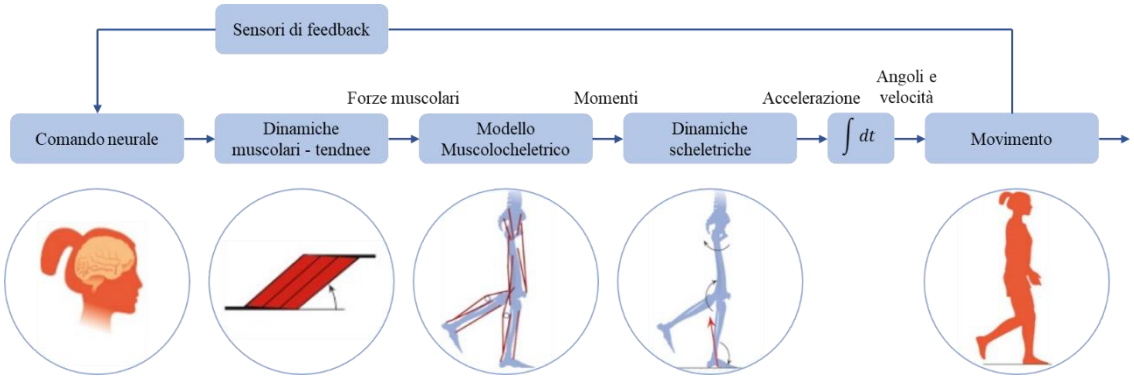

Figura 2: Forward Approach. Rappresentazione del flusso di lavoro per realizzare una simulazione muscoloscheletrica partendo dal comando neurale, per arrivare al movimento richiesto passando attraverso le dinamiche multicorpo.

In ogni caso, i dati sperimentali sono necessari per sviluppare e testare i modelli di dinamica muscoloscheletrica utilizzati nelle simulazioni di movimento, nonché per valutarne la credibilità (cioè il grado con cui le simulazioni riflettono la realtà).

Un approccio di tipo Forward si presta per l'uso della teoria dei controlli per la risoluzione di problemi biomeccanici. In particolare, ipotizzando che un atto motorio (es. camminare)

possa essere scomposto in una serie di sotto task che il sistema nervoso centrale si prefigge di raggiungere (es. spostarsi ad una data velocità, limitando il costo metabolico), il complesso problema del controllo motorio può essere visto e risolto come un problema di optimal control. Dal punto di vista matematico, costruire/definire un problema di optimal control, consiste nella definizione di una funzione (costo) complessiva rappresentativa dell'atto motorio desiderato. Per la sua risoluzione possono essere utilizzati diversi algoritmi tra cui: lo scatto singolo diretto, lo scatto multiplo diretto e la collocazione diretta.
Le simulazioni per problemi di questo tipo erano inizialmente risolte in Matlab o altri linguaggi di programmazione; solo recentemente sono stati integrate con software per la simulazione biomeccanica, come ad esempio OpenSim[3]. Per permettere l'utilizzo di metodi avanzati di optimal control, nel 2017, la Stanford University ha realizzato OpenSim Moco ("optimal control muscoloscheletrico").

Nelle sezioni successive verranno forniti maggiori dettagli sulla definizione di un problema di optimal control, con particolare attenzione all'implementazione propria del software OpenSim Moco (facendo riferimento a funzioni e nomenclatura specifiche, dove necessario).

1.2 Teoria dei controlli basata sull'Optimal Control

Commento sulla terminologia; il termine ottimizzazione dinamica è usato al posto di optimal control; sembra essere stato introdotto da Davy e Audu (1987) [9] *per distinguere il loro lavoro dall'approccio di ottimizzazione statica. Nell'ottimizzazione statica, che non è un approccio di optimal control, i momenti articolari netti vengono scomposti in singole forze muscolari, al contrario nell'optimal control sono discretizzati nel tempo* [10].

1.2.1 Introduzione all'optimal control e agli algoritmi di risoluzione

Tramite approccio di optimal control il sistema neuro-muscoloscheletrico è modellato con equazioni differenziali ordinarie soggette a controlli che influenzano il

[3] OpenSim: è una piattaforma software per la realizzazione di modelli muscoloscheletrici e permette di simularne il movimento nell'ambiente e l'interazione con esso.

comportamento del sistema (modello muscoloscheletrico). Tecniche basate sulla teoria dei controlli vengono utilizzate per trovare i controlli ottimali del modello muscoloscheletrico (ad esempio, eccitazioni muscolari, coppie articolari) che minimizzino o massimizzino un criterio di prestazione (cioè l'obiettivo del movimento) definito dall'utente (sotto forma di funzione matematica) per un determinato task motorio. Il sistema nervoso non utilizza i formalismi matematici della teoria del controllo per apprendere, pianificare ed eseguire i movimenti. Tuttavia, i metodi di optimal control possono essere utilizzati per predire il comportamento motorio volontario per una varietà di compiti di movimento, tra cui camminare, correre, saltare, stare in piedi e allungarsi.
Dal punto di vista matematico un problema di optimal control è caratterizzato da quattro elementi chiave: equazioni di stato che descrivono il comportamento dinamico del sistema in esame, controlli che possono influenzare il comportamento del sistema, vincoli che la soluzione deve soddisfare e un criterio di prestazione che dovrebbe essere ridotto al minimo (o massimizzato) mediante un'appropriata selezione dei controlli. Il vettore degli stati, in un sistema con *n* stati, è definito come:

$$\boldsymbol{y}(t) = (\boldsymbol{y}(t), \boldsymbol{y_2}(t), \dots, \boldsymbol{y_n}(t)) \quad (1.1)$$

Il vettore dei controlli, in un sistema con *m* controlli, è definito da:

$$\boldsymbol{x}(t) = (\boldsymbol{x_1}(t), \boldsymbol{x_2}(t), \dots, \boldsymbol{x_m}(t)) \quad (1.2)$$

L'equazione di stato rappresentante il comportamento del sistema sotto forma di equazione differenziale è definita come:

$$\dot{\boldsymbol{y}}(t) = \boldsymbol{f}(\boldsymbol{y}(t), \boldsymbol{x}(t), t) \quad (1.3)$$

Il problema di optimal control può essere formulato come: trovare gli stati y(t) e i controlli x(t) che minimizzano un criterio di prestazione.

$$J = \boldsymbol{\Omega}\big(\boldsymbol{y}\big(t_f\big), t_f\big) + \int_0^{t_f} \boldsymbol{\Psi}(\boldsymbol{y}(t), \boldsymbol{x}(t), t)\, dt \quad (1.4)$$

dove:

- t_f: tempo finale,

- **Ω**: è una funzione scalare (ad es. potrebbe rappresentare quantità come la velocità di decollo verticale del centro di massa in un salto verticale o l'errore nel raggiungere un bersaglio) valutata al tempo finale,
- **Ψ**: è una funzione scalare (ad es. potrebbe rappresentare la deviazione da una traiettoria di riferimento o una misura dello sforzo, come attivazione o energia metabolica, per eseguire il compito integrate sull'intero periodo di tempo.

I primi studi sul movimento umano che utilizzavano questo approccio iniziarono ad apparire in letteratura negli anni 70 [11]; [12]; [13]. Le prime applicazioni dei metodi di optimal control prevedevano algoritmi di risoluzione basati sul *principio di ottimalità* di Bellman [14], che decompone il problema in un insieme di sotto-problemi di minore dimensione ed esprime il costo totale come una somma di costi elementari associati ad ogni singola decisione. Alternativamente, venivano impiegati metodi indiretti basati sul *principio del massimo* di Pontryagin [15], il quale afferma che la funzione obiettivo debba necessariamente raggiungere un estremo tra tutti i controlli ammissibili.
A partire dagli anni '90 venne introdotto un nuovo metodo di risoluzione, attraverso il quale le variabili di controllo venivano parametrizzate e il problema di optimal control originale veniva convertito in un problema di ottimizzazione dei parametri standard [16]. Tale metodo, noto come metodo dello *scatto diretto* richiede che le equazioni di stato vengano ripetutamente integrate in avanti nel tempo per valutare il criterio di prestazione e i vincoli; inoltre, lo stato finale del sistema può essere molto sensibile a piccoli cambiamenti nelle condizioni e nei controlli iniziali, pertanto può risultare computazionalmente pesante.
Lo *scatto multiplo diretto* [17] è un altro metodo di calcolo che utilizza l'integrazione numerica a intervalli di tempo e permette di simulare intervalli consecutivi di una traiettoria imponendo che i punti finali degli intervalli adiacenti siano coerenti [18], [19]. Infine, la *collocazione diretta* prevede che i controlli e gli stati siano parametrizzati e trattati come incognite in un problema non lineare su larga scala. Quest'ultimo metodo risulta più efficiente dal punto di vista computazionale rispetto allo scatto diretto per due motivi principali. In primo luogo, non vi è alcuna integrazione numerica delle equazioni di stato, dispendiosa in termini di tempo. Al contrario, la dinamica del sistema è introdotta tramite un ampio insieme di vincoli di uguaglianza algebrica. In secondo luogo, la formulazione della collocazione diretta risulta in una matrice Jacobiana di vincoli sparsi

che consente l'uso di tecniche di algebra lineare computazionalmente efficienti. Un problema di collocazione diretta solitamente presenta più incognite rispetto a un problema comparabile di scatto diretto; tuttavia, l'implementazione della collocazione diretta può essere risolta in un tempo considerevolmente inferiore; infatti recentemente, gli approcci di collocazione diretta hanno visto un uso crescente nelle applicazioni del movimento umano [17]; [20].

1.2.2 Collocazione diretta

Il nome collocazione diretta nasce dal fatto che le derivate spline sono "collocate" con le derivate esatte [21]; [22], mentre il termine "diretta" indica il metodo usato, cioè quello diretto, la cui caratteristica fondamentale è che discretizza lo stesso problema di ottimizzazione della traiettoria convertendolo in un programma non lineare. Questo processo di conversione è noto come trascrizione, per questo che ci si può riferire ai metodi di collocazione diretta come metodi di trascrizione diretta.
In generale, i metodi di trascrizione diretta sono in grado di discretizzare un problema di ottimizzazione della traiettoria continua, approssimando tutte le funzioni continue nell'enunciato del problema come spline polinomiali su una mesh di punti temporali.
Con il termine spline ci si riferisce ad una funzione costituita da una sequenza di segmenti polinomiali. L'utilizzo dei polinomi è fondamentale perché hanno due importanti proprietà: possono essere rappresentati da un piccolo insieme finito di coefficienti ed è facile calcolarne integrali e derivate.
La dinamica viene applicata richiedendo uguaglianza tra derivata temporale delle spline e quelle dalle equazioni differenziali del sistema in punti temporali specificati. Questo produce un programma non lineare in cui gli stati sono introdotti come variabili e le dinamiche del sistema sono imposte come vincoli [23].
Metodi di questo tipo sono ancora poco diffusi e spesso sono risolti da routine sviluppate *in house* da un gruppo di ricerca, cosa che ne limita la diffusione. Da qualche anno, però, è disponibile sulla piattaforma SimTK un software open source (OpenSim Moco [24]), nato come estensione del software OpenSim [25]) che consente di realizzare problemi di optimal control risolvibili con collocazione diretta.
L'ottimizzatore di Moco approssima gli stati e i controlli come spline polinomiali su una mesh e ne risolve i punti nodali in modo che la spline obbedisca alla dinamica del sistema.

Capitolo 2: OpenSim Moco

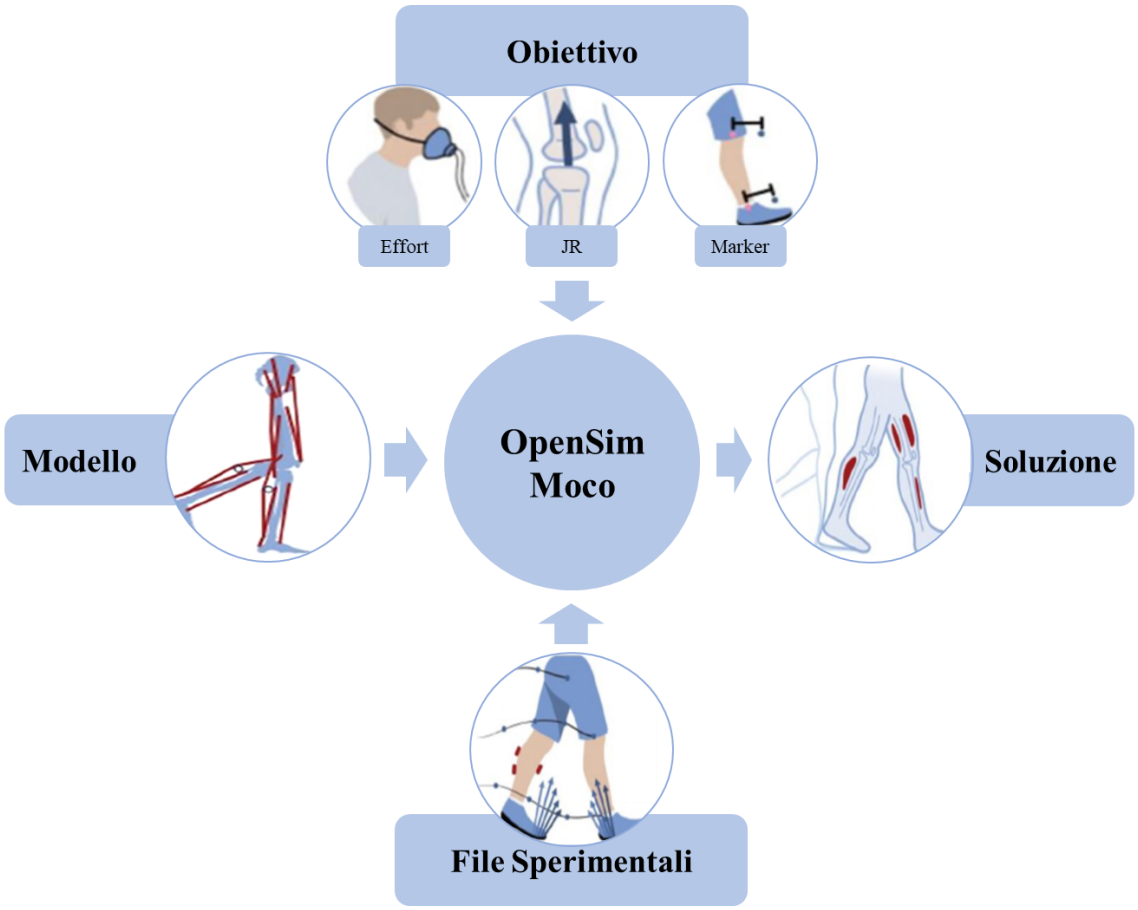

Figura 3: Struttura generale di OpenSim Moco. Mostra gli input necessari (modello) e possibili (file sperimentali e obiettivi) da inserire nel software per ottenere una soluzione.

OpenSim Moco [24] garantisce i vantaggi presenti in OpenSim (implementazione automatica delle equazioni di movimento), consente di applicare la collocazione diretta per la risoluzione dei problemi e fornisce un'interfaccia che estrae i dettagli della costruzione di un problema di optimal control. Inoltre, Moco permette la definizione di termini costo o vincoli personalizzati, a completamento/integrazione di quelli messi a disposizione.

Con OpenSim Moco è possibile fornire diversi input di dati al problema in base alla simulazione che si vuole eseguire, come illustrato in Figura 3 e meglio dettagliato nelle sezioni seguenti.

2.1 Collocazione diretta in OpenSim Moco

Moco permette di utilizzare due metodi di collocazione diretta: collocazione trapezoidale e collocazione Hermite-Simpson [26]; [23].

2.1.1 Trascrizione trapezoidale

La trascrizione trapezoidale permette di discretizzate le variabili continue in un insieme di n + 1 punti di mesh, portandolo a n intervalli di mesh. Combinando la discretizzazione con la regola trapezoidale si ottiene il seguente problema di ottimizzazione non lineare a dimensione finita da minimizzare (o programma non lineare, abbreviato in NLP):

$$\sum_j w_j J_j\left(t_0, t_f, y_0, y_n, x_0, x_n, \lambda_0, \lambda_n, p, S_{c,j}\right) + w_\lambda \sum_{i=1}^{n} trap_i(|\lambda|_2^2) \quad (2.1)$$

2.1.2 Trascrizione Hermite-Simpson

Con la trascrizione Hermite-Simpson, si hanno n + 1 punti mesh e n intervalli di mesh come nella trascrizione trapezoidale, ma vengono introdotti anche i punti medi dell'intervallo di mesh come punti di collocazione aggiuntivi. Questo porta ad un totale di 2n + 1 punti della griglia su cui vengono discretizzate le variabili continue. La NLP, da minimizzare, a dimensione finita risultante è la seguente:

$$\sum_j w_j J_{c,j}\left(t_0, t_f, y_0, y_n, x_0, x_n, \lambda_0, \lambda_n, p, S_{c,j}\right) + w_\lambda \sum_{i=1}^{n} simpson_i(|\lambda|_2^2) \quad (2.2)$$

2.2 Problemi di optimal control

I problemi di optimal control presenti in OpenSim Moco sono caratterizzati da tre elementi matematici fondamentali: le variabili di stato (posizioni e velocità degli stati), dai controlli (momenti articolari, forze muscolari, eccitazioni muscolari o eccitazioni neurali) e da un criterio di prestazione, cioè una funzione costo da massimizzare o minimizzare.

Un problema di optimal control viene risolto, dal punto di vista matematico, tramite la minimizzazione di una funzione costo espressa come:

$$\sum_j w_j \, J_j\left(t_0, t_f, y_0, y_f, x_0, x_f, \lambda_0, \lambda_f, p, S_{c,j}\right) \quad (2.3)$$

L'obiettivo finale del problema è determinare gli stati ($y(t)$) e i controlli ($x(t)$) che minimizzano la funzione costo (J), data dalla somma di funzioni costo pesate (w_j)(J_j) (i termini peso indicano l'importanza di ridurre al minimo una specifica funzione costo), anche se soggetti alle dinamiche multicorpo (che comprendono la matrice di massa M, le forze gravitazionali, muscolari e di altro tipo applicate (F_{app}), forze inerziali ($F_{inerziali}$) e dinamiche ausiliarie (esprimibili come equazioni differenziali esplicite $f_{z,ex}$ o implicite $f_{z,im}$)). Le funzioni di costo dipendono da limiti temporali (t_0, t_f), dagli stati (y_0, y_f), dai controlli (x_0, x_f), dai moltiplicatori di Lagrange (λ_0, λ_f), dai parametri tempo invarianti e integrali ($S_{b,k}$). Le variabili presentano un limite inferiore (L) e superiore (U), mentre gli stati e i controlli hanno limiti iniziali e finali.
È possibile che siano presenti vincoli cinematici, causati dalle forze applicate da parti del sistema modellato. Per risolverli si utilizzano i moltiplicatori di Lagrange (λ), variabili nel tempo. Derivando i vincoli cinematici (φ) si introduce il vincolo cinematico Jacobiano (G). Sono infine presenti anche vincoli di confine (V_k) e di percorso (g) con i corrispondenti limiti inferiore e superiore.

OpenSim Moco consente all'utente di risolvere un problema di optimal control inserendo e/o combinando diverse caratteristiche, come ad esempio: vincoli cinematici, dinamiche multicorpo implicite, dinamiche ausiliarie implicite e cinematica prescritta.

2.2.1 Vincoli cinematici

Il supporto per problemi che incorporano vincoli cinematici (ad es. il moto della patella viene vincolato a quello del ginocchio) è una caratteristica chiave di Moco.

La risoluzione di un problema di dinamica muscoloscheletrica seguendo un approccio di tipo diretto (forward approach) richiede che le coordinate e le velocità generalizzate iniziali soddisfino alcuni vincoli cinematici (es. $\varphi(q) = 0$) e che la loro derivata prima

sia nulla ($\dot{\varphi}(q,u) = 0$). Durante l'integrazione vengono risolte le accelerazioni generalizzate e i moltiplicatori di Lagrange che soddisfano le equazioni del moto, della dinamica multicorpo, e la derivata seconda dei vincoli cinematici, $\ddot{\varphi}(q,u,\dot{u}) = 0$. Integrando numericamente le accelerazioni generalizzate risultanti si ottengono coordinate e velocità generalizzate che giacciono sullo spazio dei vincoli del sistema definito da $\varphi(q) = 0$ (cioè la superficie multidimensionale su cui sono soddisfatti i vincoli cinematici). Per correggere eventuali errori nei vincoli causati dall'errore di integrazione numerica, le coordinate e le velocità generalizzate vengono proiettate sullo spazio dei vincoli [27].

Nella collocazione diretta, viene risolta contemporaneamente l'intera traiettoria del sistema, comprese le coordinate generalizzate, le velocità generalizzate e i moltiplicatori di Lagrange. La gestione dei vincoli richiede un approccio differente poiché non è possibile eseguire una "proiezione" interna all'interno del problema di ottimizzazione della collocazione. Per questo viene usato il metodo sviluppato da Posa et al. [28] in cui il risolutore di ottimizzazione ricerca i moltiplicatori di Lagrange che soddisfano le equazioni.

2.2.2 Dinamiche multicorpo implicite

Quando si applica la dinamica multicorpo [27] (equazioni della dinamica che governano il sistema) tramite un'equazione differenziale implicita, $\dot{u}$ risulta una variabile del problema di optimal control e l'ottimizzatore ricerca il valore che ne soddisfa l'equazione della dinamica nella sua forma originale. Questo approccio risulta simile al calcolo della "dinamica inversa", in cui si conosce $\dot{u}$ e si risolve l'equazione della dinamica multicorpo per ottenere le forze generalizzate necessarie.

Simbody fornisce un operatore di "dinamica inversa" per calcolare la forza residua (f_{residual}), utilizzabile in un vincolo di percorso per rendere implicita la dinamica multicorpo. Le accelerazioni calcolate differiscono da quelle dell'ottimizzatore, per questo sono utilizzati dei moltiplicatori di Lagrange di movimento (λ_m) impostati per essere le forze richieste per la dinamica multicorpo da mantenere.

$$M\dot{u} + f_{\text{inertial}} + \lambda_m = f_{\text{app}} \quad (2.4)$$

Nel caso di vincoli cinematici, le equazioni del moto contengono sia i moltiplicatori di Lagrange di vincolo cinematico λ_c che i moltiplicatori di Lagrange di movimento λ_m:

$$M(q)\dot{u} + f_{inertial}(q,u) + G(q)^T\lambda_c + \lambda_m = f_{app}(t,q,u,z,x) \quad (2.5)$$

2.2.3 Dinamiche ausiliarie implicite

Al contrario della dinamica multicorpo implicita che risiede nel solutore, la dinamica implicita ausiliaria (ad es. dinamiche di attivazione muscolare e dinamiche di cedevolezze tendinee) risiede nel problema e nei componenti del modello. Questa differenza esiste per due motivi: (i) le equazioni per la dinamica multicorpo esistono al di fuori del modello, mentre le equazioni per la dinamica ausiliaria esistono all'interno del modello e (ii) gli utenti potrebbero desiderare che solo un sottoinsieme di stati venga gestito implicitamente, mentre la scelta della modalità dinamica per il sistema multicorpo influisce su tutte le equazioni differenziali che descrivono il sistema multicorpo.
La gestione della dinamica implicita richiede che Moco fornisca ai componenti i valori per le derivate delle variabili di stato ausiliarie e ottenga i residui delle equazioni differenziali implicite, Per questo i componenti devono essere implementati con *Component::computeStateVariableDerivatives()* così da fornire la forma esplicita delle equazioni differenziali; questo è ancora vero per i componenti che supportano una modalità implicita. I componenti che supportano la modalità implicita devono supportare anche la modalità esplicita in modo che i modelli utilizzati nei problemi di collocazione diretta possano ancora essere utilizzati nelle simulazioni con forward approach. La corretta implementazione di *computeStateVariableDerivatives()* in modalità implicita consiste nell'impostare la derivata sul valore della variabile discreta contenente la derivata di stato fornita da Moco.

2.2.4 Cinematica prescritta

OpenSim Moco permette di imporre diversi input, come ad esempio la cinematica prescritta che consiste nello stimare il comportamento del muscolo e dell’attuatore che ha guidato un movimento prescritto precedentemente.

Tramite Simbody[4] è possibile prescrivere il valore di una coordinata in due modi:

1. aggiungendo un vincolo con *SimTK::Constraint::PrescribedMotion* (mostra le proprietà delle funzioni e coordinate prescritte).
2. rimuovendo un grado di libertà con *SimTK::Motion*.

L'aggiunta di un vincolo porta ad aumentare il numero di equazioni del sistema mantiene la dinamica multicorpo, mentre la rimozione di un grado di libertà permette di ottenere un sistema di equazioni più ristretto.
Moco utilizza la componente *PositionMotion* (che utilizza *SimTK::Motion*) per la prescrizione del movimento di tutti i gradi di libertà attraverso le funzioni spline presenti nel problema di collocazione diretta, ma impedisce di prevedere le deviazioni cinematiche dal movimento osservato.

In questa formulazione da minimizzare, le variabili cinematiche q e u sono sostituite con quantità note $\hat{q}$ e $\hat{u}$:

$$\sum_j w_j J_j\left(t_0, t_f, \widehat{q_0}, \widehat{q_f}, \widehat{u_0}, \widehat{u_f}, z_0, z_f, x_0, x_f, \lambda_0, \lambda_f, p, S_{c,j}\right) \quad (2.6)$$

Il sistema contiene ancora variabili di stato ausiliarie (*z*), variabili di controllo (*x*) e dinamica ausiliaria. Se nessuna delle variabili dei parametri influenza il sistema multicorpo, la dinamica multicorpo viene ridotta a un equilibrio di forze: le forze applicate devono corrispondere alle forze generalizzate nette determinate dalla cinematica (cioè la dinamica inversa).
Indipendentemente dal fatto che il movimento sia prescritto aggiungendo vincoli o eliminando variabili, OpenSim integra gli elementi di forza modellati con moltiplicatori di Lagrange per garantire il raggiungimento del movimento prescritto. Quando si utilizza *PositionMotion* con Moco, è necessario che i moltiplicatori di Lagrange del movimento prescritto siano pari a zero, garantendo così che il movimento sia completamente generato dagli elementi di forza modellati.

[4] Simbody è una libreria C++ open source ad alte prestazioni di livello industriale che fornisce un trattamento sofisticato di sistemi multicorpo articolati con particolare attenzione alle esigenze della simulazione biomedica. È utile per simulazioni dinamiche predittive di diversi sistemi biologici come modelli biomeccanici neuromuscolari di OpenSim.

2.3 Struttura delle classi presenti in Opensim Moco

Moco utilizza una libreria a moduli di costo e vincolo, implementata con classi software che descrivono il problema e come deve essere risolto.
La classe principale è il *MocoStudy*, che è a sua volta diviso in due classi: *MocoProblem* e *MocoSolver*, indipendenti l'una dall'altra. Le classi Moco sono disponibili tramite file di testo C++, MATLAB, Python e XML, con interfacce familiari agli utenti di OpenSim.

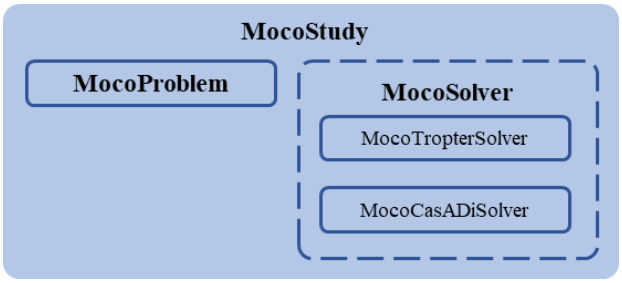

Figura 4: Struttura delle classi di OpenSim Moco. La classe principale, che racchiude le altre sottoclassi, è il MocoStudy, al suo interno è presente la classe MocoProblem e MocoSolver.

2.3.1 MocoStudy

MocoStudy rappresenta la classe di primo livello per la risoluzione di un problema di optimal control personalizzato.
Questa classe è composta da un *MocoProblem* , che descrive il problema di optimal control, e un *MocoSolver* , che descrive il metodo numerico per risolvere il problema. Quando si crea un *MocoStudy* a livello di codice, il flusso di lavoro inizia dalla costruzione del *MocoProblem* (viene impostato il modello, vincoli, goals ecc.); dopodiché è possibile inizializzare il solutore (*MocoSolver*) e modificarne le impostazioni. Infine, è possibile accedere alla soluzione del problema (*MocoSolution*). Come passaggio facoltativo è possibile visualizzare o elaborare ulteriormente la soluzione.

2.3.2 MocoProblem

La classe *MocoProblem* contiene tutti gli elementi che descrivono un problema di optimal control ed è un elenco di fasi (*MocoPhase*) che permettono di definire il problema, ma non consentono di risolverlo (contiene solo l'input dell'utente).

La *MocoPhase* iniziale è costituita dal modello muscoloscheletrico di OpenSim usato per generare un sistema di equazioni algebriche differenziali che descrive stati, controlli, vincoli cinematici, dinamiche multicorpo ausiliarie (muscolari, cedevolezza tendinea, ecc). Ad ogni passo/iterazione, la *MocoPhase* viene/può essere modificata agendo su alcuni suoi parametri (*MocoParameters*), es. la massa o la lunghezza ottimale delle fibre muscolari, consentendo di risolvere il problema. L'obiettivo da raggiungere, cioè la funzione costo da ridurre al minimo (es. la somma ponderata di controlli al quadrato) o i vincoli di endpoint da applicare è invece descritta dalla *phase MocoGoals*. I vincoli algebrici non cinematici da applicare all'intera fase sono descritti dai cosiddetti *MocoPathContraints*. In generale, è possibile vincolare qualsiasi funzione del tempo a trovarsi in un intervallo specificato, ad esempio è possibile vincolare le eccitazioni muscolari simulate per avvicinarsi a quelle sperimentali da elettromiografia. Infine, è possibile creare limiti iniziali e finali sul tempo, variabili di stato e controllo iniziale e finale.

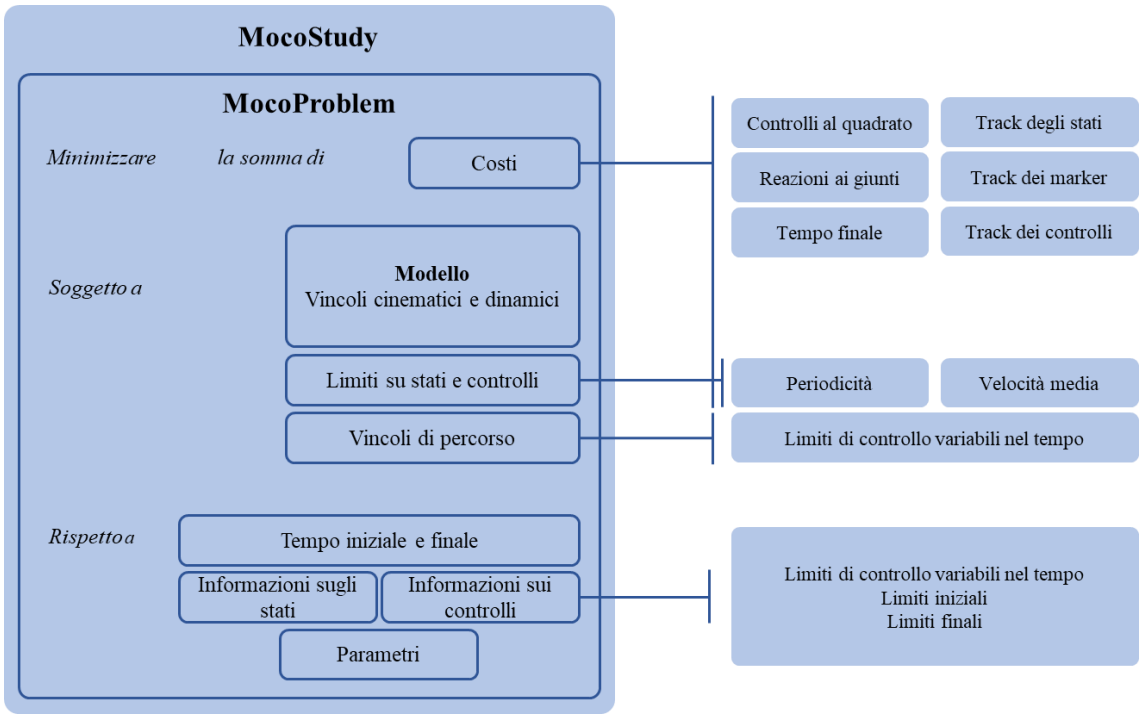

Figura 5: Struttura del MocoProblem. Mostra la struttura di primo livello,MocoStudy, che al suo interno contiene il MocoProblem, nel quale sono contenuti tutti gli elementi o Phase per realizzare un problema di optimal control.

2.3.4 MocoSolver

La classe *MocoProblem* contiene solo l'input dell'utente e descrive il problema di optimal control fisico. Le classi *MocoSolver* consentono all'utente di configurare il metodo numerico utilizzato per risolvere il problema.

Esistono due tipologie di solutore disponibili in Moco: *MocoTropterSolver* e *MocoCasADiSolver* [29], che utilizzano solutori open source come IPOPT [30] e SNOPT [31].

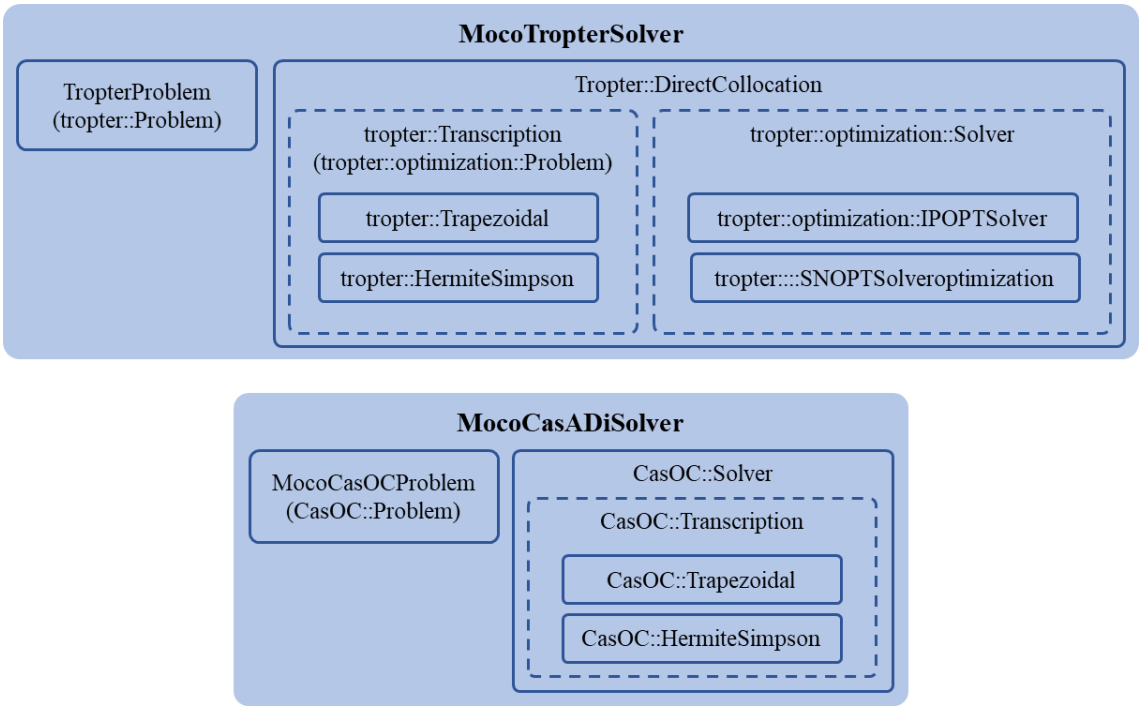

Figura 6: Struttura delle classi per i risolutori numerici di OpenSim Moco. Due tipologie di solutori possibili: Tropter e CasADi, entrambi caratterizzati dalla loro struttura interna.

Il solutore Tropter contiene una coppia di risolutori per problemi di optimal control: *TropterProblem* (che deriva da *tropter::Problem*) e *DirectCollocation*. Per risolvere il problema di controllo ottimo il solutore *DirectCollocation* usa una coppia problem-solver per l'ottimizzazione non lineare generica costituita da uno schema di trascrizione (che deriva da *tropter::optimization::Problem*) e da un solutore (che deriva da *tropter::optimization::Solver*).

Il solutore CasADi utilizza anch'esso una coppia problem-solver costituita da un problema (*MocoCasOCProblem* derivante da *CasOC::Problem*) e da un solutore (*CasOC::Solver*).
Queste coppie di classi consentono una separazione tra il modo in cui un utente specifica un problema e il metodo utilizzato per risolverlo.

I passaggi fondamentali per l'utilizzo del solutore CasADi iniziano con la risoluzione del *MocoProblem* con le impostazioni del *MocoSolver* richieste precedentemente. Dopo aver inizializzato le varie *phase* del *MocoProblem* (obiettivi, vincoli di percorso, parametri, ecc.) viene creata un'istanza del problema (*CasOC::Problem***)** in cui vengono aggiunte le *phase* inizializzate in precedenza. Le impostazioni del risolutore sono copiate in un'istanza del solutore (*CasOC::Solver***)**. Dopo aver creato uno schema di trascrizione (*CasOC::Trapezoidal* o *CasOC::HermiteSimpson*) che permette di convertire l'istanza del problema (*CasOC::Problem*) in un programma non lineare. Viene creata una funzione CasADi in grado di risolverlo (*casadi::nlpsol()*). Al termine *MocoStudy* restituisce una *MocoSolution* che viene utilizzata come ipotesi iniziale di *MocoTrajectory*[5] nell'iterazione successiva. Il *MocoTropterSolver* segue passaggi simili.

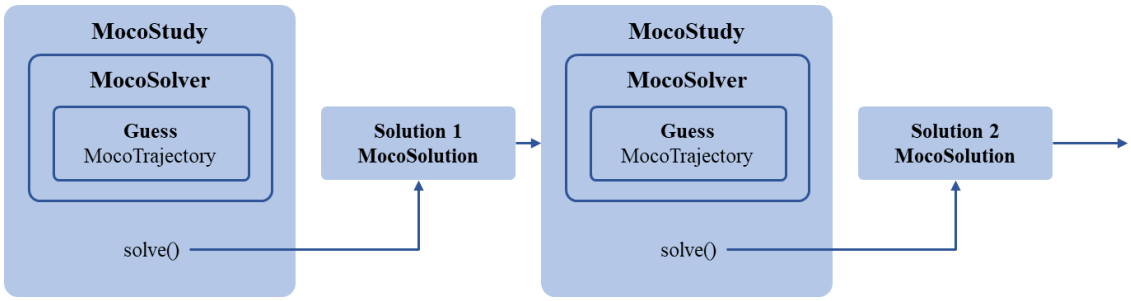

Figura. 7: Modello iterativo delle soluzioni generate da Moco durante il processo di ottimizzazione. Le ipotesi per l'ottimizzazione vengono specificate utilizzando MocoTrajectory, contenente i valori degli stati, controlli, moltiplicatori di Lagrange e parametri a qualsiasi iterazione nell'ottimizzazione. MocoSolution è una sottoclasse di MocoTrajectory contenente la soluzione di uno studio, lo stato di successo dell'ottimizzazione, il valore obiettivo finale e il numero di iterazioni del risolutore.

[5] MocoTrajectory rappresenta la classe che contiene i valori delle variabili di un problema di controllo ottimale, può essere utilizzata per specificare un'ipotesi iniziale o mantenere la soluzione restituita da un risolutore.

2.4 Tools di OpenSim Moco

Oltre ad un generico *MocoStudy* che l'utente può creare e modificare secondo le proprie esigenze, OpenSim Moco fornisce due tools preimpostati: *MocoInverse* e *MocoTrack*. La struttura con cui sono realizzati è analoga a quella descritta precedentemente, ma le funzioni costo sono predefinite (pur consentendo un margine di adattamento) e sono tali da eseguire delle analisi sostanzialmente simili a quelle tipicamente implementate nei workflow più tradizionali (approccio inverso [8], [25] e EMG-assisted [32], [33]). Nelle sezioni successive i singoli Tool vengono descritti più in dettaglio.

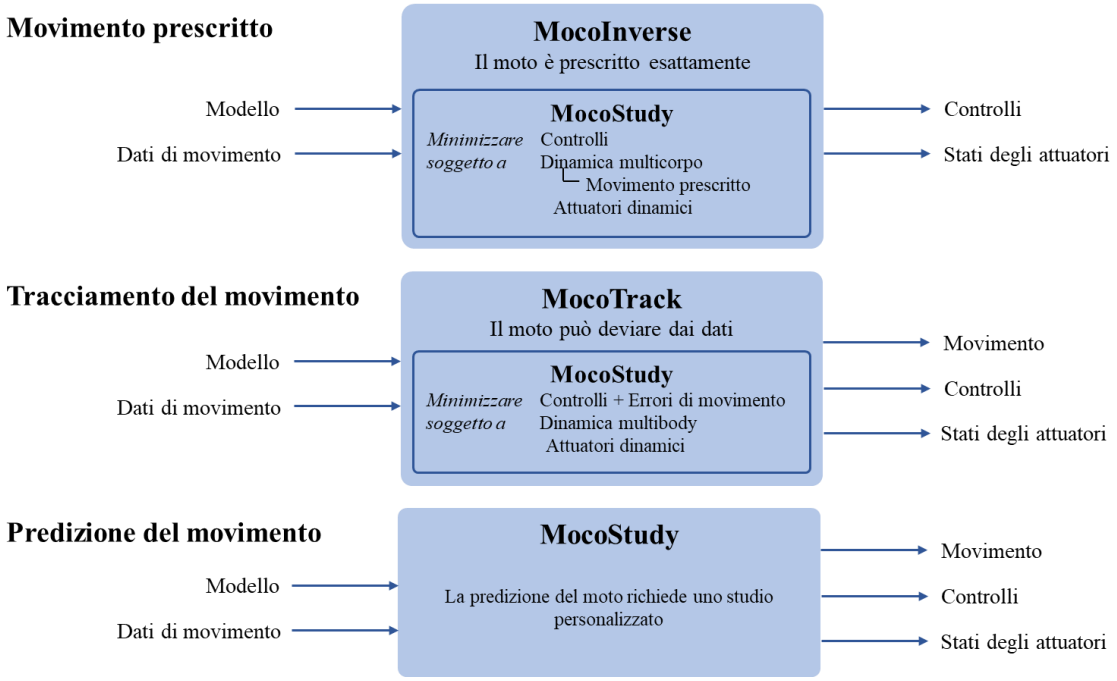

Figura 8: Schema generale rappresentante la struttura generale dei vari tools (MocoInverse, MocoTrack e MocoStudy personalizzaabile, cioè MocoPredict) per la risoluzione di problemi di movimento prescritto, tracciamento del movimento e predizione del movimento.

Come accennato, il software OpenSim Moco consente all'utente di generare il proprio studio (*MocoStudy*) personalizzato. È possibile, ad es. realizzare simulazioni predittive (*MocoPredict*) mediate la creazione di una funzione costo (rappresentativa dell'atto motorio) costituita dall'unione di più sottofunzioni. Per completezza, si fa notare che le tre categorie di simulazione (*MocoInverse*, *MocoTrack* e *MocoPredict*) presentano tempi computazionali crescenti.

2.4.1 MocoInverse

MocoInverse è uno strumento che serve per risolvere problemi di ottimizzazione in cui è nota la cinematica, tra cui il problema della ridondanza muscolare [34]. Attraverso l'aggiunta di una componente al modello (*PositionMotion*), l'utente può fornire (imporre) una cinematica realistica, es. basata su dati sperimentali. Il vantaggio di avere il movimento prescritto è che il problema è più robusto e si risolve più velocemente, poiché la dinamica multicorpo non lineare non fa più parte del problema di ottimizzazione. Lo svantaggio è che non risulta possibile prevedere deviazioni dal movimento osservato, in quanto quest'ultimo è imposto.
Per assicurarsi che la cinematica fornita, generata attraverso curve di tipo spline, soddisfino i vincoli cinematici, *MocoInverse* opera piccole modifiche sul dato in ingresso (*Model::assemble()*) dove necessario. Ciò nonostante, non risulta sempre possibile riprodurre il dato sperimentale.
Ci sono più possibili cause per questo:

- i muscoli non sono abbastanza forti per raggiungere i momenti articolari netti richiesti,
- i momenti articolari netti cambiano più rapidamente di quanto consentano le costanti di tempo di attivazione e disattivazione,
- il filtraggio dei dati provoca momenti congiunti netti desiderati non realistici.

Per questi motivi, è necessario aggiungere attuatori di "riserva" al modello in modo tale che la conservazione della quantità di moto sia soddisfatta (analogamente a quanto viene fatto in OpenSim).

MocoInverse può essere paragonato alla Static Optimization, in quanto in entrambi i casi vengono utilizzano dati di cinematica per minimizzare una funzione costo per trovare la minima forza muscolare per compiere la cinematica fornita. La differenza sta che *MocoInverse* sfrutta la collocazione diretta per la risoluzione del problema, quindi, analizza tutta la traiettoria per ottenere l'ottimizzazione, mentre nella Static Optimization le attivazioni e le forze muscolari vengono risolte separatamente in ogni istante di tempo (indipendentemente dalla soluzione trovata per l'istante precedente e successivo) [35].

2.4.2 MocoTrack

In Moco, è possibile utilizzare lo strumento *MocoTrack* per risolvere i problemi di ottimizzazione del tracciamento. In questo caso, attraverso l'implementazione di una funzione costo rappresentativa dell'obiettivo finale da raggiungere, e composta di due termini di tracciamento, il software cercherà la soluzione in grado di seguire quanto più possibile il dato sperimentale fornito. I due principali (sotto)obiettivi inclusi nella funzione costo del tool *MocoTrack* sono: *MocoMarkerTrackingGoal* e/o *MocoStateTrackingGoal*. Entrambi i tipi di dati cinematici possono essere tracciati contemporaneamente e possono essere forniti pesi per ogni termine di costo. Inoltre, i pesi possono essere impostati per i singoli componenti dei set di dati tracciati (ad esempio, tracciando i marcatori sui punti di riferimento ossei preferenzialmente rispetto ai marcatori sui tessuti molli).

Ulteriori funzionalità fornite da *MocoTrack* includono garantire che la finestra temporale del problema sia coerente con tutti i set di dati, compilare gli stati di velocità delle coordinate mancanti per monitorare se sono disponibili solo i valori delle coordinate e aggiungere i dati di rilevamento all'ipotesi iniziale del problema.

MocoTrack è utile per prevedere le deviazioni dai dati di movimento e può utilizzare modelli di contatto, cioè l'aggiunta di sfere al modello che simulino il contatto con una superficie; questo permette di realizzare un tracciamento delle forze di reazione al terreno. Tramite l'uso di attivazioni dinamiche nel modello, *MocoTrack* è in grado di fornire un'ipotetica attivazione muscolare rappresentativa del movimento tracciato. Questa funzione può essere implementata ulteriormente con un tracciamento dei dati EMG per migliorare la corrispondenza fisiologica. In questo senso, il funzionamento del *MocoTrack* è assimilabile all'approccio EMG-assisted [32]; [33]: le eccitazioni ed attivazioni muscolari sono stimate cercando di replicare con un certo grado di fedeltà gli inviluppi lineari dei segnali EMG raccolti in vivo, cercando di soddisfare al meglio le equazioni di equilibrio dinamico del sistema. Il risultato è la migliore corrispondenza possibile tra i modelli di attivazione muscolare osservati fisiologicamente e le forze previste, pur soddisfacendo i vincoli di momento su tutti e tre gli assi articolari.

2.4.3 MocoPredict

MocoPredict non rappresenta effettivamente un vero e proprio tool, ma è uno studio completamente personalizzato che permette di determinare la cinematica articolare, i segnali di attivazione, i controlli necessari, ecc. che devono essere applicati ad un modello per garantire la miglior prestazione del task motorio desiderato.
Nel caso del *MocoPredict* (come per un *MocoStudy* in generale), in input viene fornito solamente il modello muscoloscheletrico, ma non dati sperimentali. L'utente può modificare a piacimento la funzione costo, aggiungendo vincoli, controlli e/o obiettivi di endpoint in modo tale da eseguire varie simulazioni predittive.

2.5 MocoGoals

OpenSim Moco permette l'utilizzo di una serie di funzioni costo già realizzate per l'ottimizzazione di un determinato task motorio prescritto o predetto.
Di seguito verrà riportata una panoramica delle funzioni costo implementabili, limitandosi a quelle più comuni, per i quali verrà fornita una breve definizione. Per una lista più dettagliata, il lettore è rimandato alla guida di OpenSim Moco (DoxyGen[6]).

2.5.1 MocoControlGoal

La funzione MocoControlGoal consente di minimizzare la somma del valore assoluto dei controlli elevato ad un dato esponente, integrato sulla fase.

$$J = \frac{1}{d}\int_{t_i}^{t_f} \sum_{c \in C} w_c |x_c(t)|^p \, dt \quad (2.7)$$

dove:

- t_i: tempo iniziale
- t_f: tempo finale
- d: distanza percorsa dal sistema
- C: insieme del set di controlli

[6] https://opensim-org.github.io/opensim-moco-site/docs/

- w_c: peso del controllo *c*-esimo
- $x_c(t)$: segnale di controllo *c*-esimo
- p: esponente

2.5.2 MocoMarkerTrackingGoal

La funzione *MocoMarkerTrackingGoal* permette di tracciare le traiettorie di markers sperimentali, attraverso la minimizzazione della sommatoria degli scarti quadratici tra coppie di markers corrispondenti (sul file sperimentale e sul modello). L'utente può attribuire un peso diverso ai vari markers virtuali (sul modello) per dare più o meno importanza agli errori individuali.

$$J = \sum_{k=1}^{n} \left(m_{S_k} - w_k \cdot m_{M_k}\right)^2 \quad (2.8)$$

dove:

- w_k: peso del marker k-esimo del modello
- m_{S_k}: posizione del marker k-esimo sperimentale
- m_{M_k}: posizione del marker k-esimo del modello

2.5.3 MocoStateTrackingGoal

La funzione *MocoStateTrackingGoal* permette di minimizzare la somma degli scarti quadratici tra coppie di variabili di stato corrispondenti (sul file sperimentale e sul modello). L'utente può attribuire un peso diverso alle varie variabili per dare più o meno importanza agli errori individuali. Questa funzione è fondamentale per realizzare un tracciamento basato su cinematica sperimentale.

$$J = \int_{t_i}^{t_f} \sum_{k=1}^{n} \left(x(t)_{rif_k} - w_k \cdot x(t)_k\right)^2 \quad (2.9)$$

dove:

- t_i: tempo iniziale
- t_f: tempo finale

- w_k: peso dello stato k-esimo
- $x(t)_k$: variabile di stato k-esima
- $x(t)_{rif_k}$: variabile di stato k-esima di riferimento

2.5.4 MocoPeriodicityGoal

Questo obiettivo impone l'uguaglianza tra i valori delle variabili iniziali e finali nel problema di controllo ottimo, a simulare una periodicità nel movimento (es. camminata su tapis roulant). I valori iniziale e finale possono appartenere a variabili continue separate o alla stessa variabile continua, purché siano dello stesso tipo di variabile (es. stato o controllo). Le coppie di valori vengono specificate tramite un *MocoPeriodicityGoalPair*, dove la variabile iniziale della coppia indica la variabile di stato/controllo iniziale e la variabile finale indica la variabile di stato/controllo finale. Impostare le variabili iniziale e finale sulla stessa variabile per imporre la periodicità su una singola variabile continua. Sono supportate solo le coppie di variabili continue state e control, specificate tramite le proprietà *'state_pairs'* e *'control_pairs'*.
Un *MocoPeriodicityGoal* può essere utilizzato per simulare mezzo ciclo del passo o un ciclo completo e ripeterlo per realizzare un passo completo o più passi.

2.5.5 MocoJointReactionGoal

La funzione *MocoJointReactionGoal* è utilizzata per minimizzare la somma dei quadrati dei momenti di reazione e dalle misure di forza di un dato giunto, integrati sulla fase.

2.5.6 MocoAverageSpeedGoal

Questo obiettivo richiede che la velocità media del sistema corrisponda alla velocità media desiderata. La velocità media del sistema è lo spostamento del centro di massa del sistema diviso per la durata della fase.
In modalità vincolo endpoint, l'obiettivo viene calcolato come segue:

$$v_{\text{des}} - \frac{r_{\text{com}}(t_f) - r_{\text{com}}(t_i)}{t_f - t_i} \quad (2.10)$$

dove:

- $r_{\text{com}}(t)$: spostamento del centro di massa
- t_i: tempo iniziale
- t_f: tempo finale

2.5.7 MocoSumSquaredGoal

La funzione *MocoSumSquaredGoal* permette di minimizzare la somma dei quadrati degli stati (angoli articolari o attivazioni), integrati sulla fase. Ad esempio, questo può essere utilizzato per ridurre al minimo le attivazioni muscolari. L'utente ha la possibilità di personalizzare il peso relativo al singolo stato per dargli più o meno importanza.
Questo obiettivo è calcolato come segue:

$$J = \int_{t_i}^{t_f} \sum_{s \in S} w_s y_s(t)^2 dt \quad (2.11)$$

dove:

- t_i: tempo iniziale
- t_f: tempo finale
- S: set di variabili di stato usate nell'obiettivo
- w_s: peso per la variabile s
- $y_s(t)$: variabile di stato

Capitolo 3: Materiali e metodi

Nel seguente capitolo verranno esposti i materiali utilizzati e i metodi messi a punto per la realizzazione di una simulazione predittiva di camminata tramite OpenSim Moco.

3.1 Materiali

In questa prima sezione, verranno descritti uno ad uno i dati sperimentali impiegati per la conduzione dello studio, per terminare con una breve descrizione dei modelli muscoloscheletrici utilizzati.
Per semplificare la dissertazione e facilitarne la comprensione, i dati - provenienti da diverse fonti - verranno introdotti sulla base del loro utilizzo:

1. definizione della condizione iniziale
2. simulazione predittiva

3.1.1 Dati sperimentali per la condizione iniziale

Per definire la condizione iniziale per la simulazione predittiva di camminata sono stati utilizzati dati sperimentali provenienti da un dataset pubblico [36], relativo ad una popolazione adulta sana. Il dataset contiene dati acquisiti in un laboratorio di analisi del movimento, tra cui le traiettorie di 52 markers retroriflettenti posizionati su landmark anatomici (stereofotogrammetria), le forze di reazione al terreno misurate da pedane di forza e i segnali elettromiografici (dei principali muscoli degli arti inferiori) registrati mentre i soggetti eseguivano una camminata su livello piano a 5 diverse velocità (0 – $0.4ms^{-1}$, $0.4 – 0.8ms^{-1}$, $0.8 – 1.2ms^{-1}$, velocità spontanea e velocità auto-determinata) e mentre mantenevano una posa statica.
L'analisi del movimento è ottenuta tramite un sistema optoelettronico a 10 telecamere con campionamento a 100Hz (OQUS4, Qualisys, Svezia), due pedane dinamometriche con campionamento a 1500Hz (OR6-5, AMTI, USA) per la registrazione della forza e

momento di reazione al suolo e sistema elettromiografico (EMG) wireless a 8 canali con campionamento a 1500Hz (Desktop DTS, Noraxon, USA)

In questo elaborato di tesi, per la definizione della condizione iniziale, sono stati utilizzati i dati relativi ad un unico soggetto (#2014001) di altezza e peso pari a 166 cm e 67 kg, rispettivamente. In particolare, sono stati scelti i dati sperimentali relativi ad una camminata a una velocità compresa tra 0.8 e 1.2ms^{-1}, considerata rappresentativa di una camminata 'normale'. Una camminata a velocità inferiore (es. 0.4-0.8 ms^{-1}) o superiore (es. velocità spontanea), tra quelle disponibili, avrebbero verosimilmente causato problemi alla simulazione predittiva.

3.1.2 Dati sperimentali relativi alla simulazione predittiva

I dati sperimentali utilizzati per la simulazione predittiva sono stati ricavati dalla sesta edizione della "Grand challenge competition to predict in vivo knee loads" [37], [38]. Questo dataset contiene, oltre alle traiettorie di marcatori sperimentali e segnali EMG registrati mentre il soggetto eseguiva vari task motori, anche immagini tomografiche degli arti inferiori, e le registrazioni dei carichi articolari al ginocchio misurati attraverso una protesi strumentata con celle di carico.
L'analisi del movimento è ottenuta con l'uso di un sistema optoelettronico con campionamento a 100Hz, quattro pedane di forza con campionamento a 1000Hz, sistema EMG a 14 canali con campionamento a 1000Hz.
Inoltre, è presente un file di statica (con il soggetto in posizione di riferimento) per garantire la scalatura[7] del modello muscoloscheletrico. I dati in questione si riferiscono ad un soggetto anziano di peso 70kg e altezza 172 cm.

3.1.3 Modello muscoloscheletrico

Il modello muscoloscheletrico utilizzato per il progetto si basa sul Full Body Model (o modello Rajagopal 2015) [39] che include le geometrie ossee degli arti inferiori e superiori del corpo umano (rappresentativo di un soggetto maschio di 75 kg, alto 1.70 cm

[7] Scalatura: processo attraverso cui viene modificata l'antropometria di un modello al fine da meglio approssimare le dimensioni del soggetto sotto studio. Il ridimensionamento viene in genere eseguito confrontando i dati dei marcatori sperimentali con i marcatori virtuali posizionati sul modello.

[40], [41]), nonché 80 attuatori lineari (40 per lato) rappresentativi delle unità muscolo tendinee degli arti inferiori e 17 attuatori di coppia ideali per guidare la parte superiore del corpo (tre relativi ai gradi di libertà della spalla, due relativi alla rotazione e flessione del gomito e due per i gradi di libertà del polso). Il modello possiede 37 gradi di libertà che ne definiscono la cinematica articolare. In particolare, la posizione del bacino rispetto al suolo è rappresentata da tre gradi di libertà traslazionali indipendenti (tx, ty e tz), l'orientamento invece è determinato da tre rotazioni (Tilt, List, Rotation). Il femore è articolato con il bacino tramite un giunto sferico che permette il movimento in estensione, flessione, abduzione, adduzione, rotazione esterna e rotazione interna. L'articolazione del ginocchio è modellata tramite un grado di libertà con movimento di flessione ed estensione. Le articolazioni della caviglia, dell'astragalo e della metatarso-falangea sono modellate come coppie rotoidali con movimento di dorsi-flessione della caviglia, l'inversione e flessione della punta. La testa e il busto sono stati modellati come un segmento rigido collegato al bacino da un giunto sferico che permette tre rotazioni (flexion, bending e rotation). L'omero è articolato con il busto con un giunto sferico, mentre l'ulna è collegata all'omero tramite una coppia rotoidale. La pronazione dell'avambraccio è modellata come un'articolazione a perno che collega il radio e l'ulna. Un giunto cardano a due gradi di libertà collega la mano al radio.

3.1.4 Elaborazione dei dati sperimentali

I dati sperimentali, in formato C3D sono stati successivamente elaborati con MOtoNMS [42], che ha permesso di ottenere i file di input necessari per le analisi e simulazioni in OpenSim Moco.
L'elaborazione prevede il calcolo dei centri di pressione e delle coppie per le pedane di forza, la rotazione dei dati di motion capture tra diversi sistemi di coordinate (quelli delle pedane di forza e laboratorio), il filtraggio dei segnali EMG, il calcolo del massimo livello di eccitazione e la successiva normalizzazione dell'inviluppo lineare (del segnale EMG).

Il file è strutturato in modo che la frequenza massima di interpolazione delle traiettorie sia 100Hz per la condizione iniziale e 120Hz per i dati della KGC6, con frequenza di cutoff per il filtraggio pari a 8Hz. La finestra di analisi è stata fatta corrispondere ad un ciclo del passo completo, dal contatto iniziale del tallone destro con il terreno fino al

contatto successivo dello stesso piede (così da includere sia la fase di appoggio che la fase di volo). Il massimo valore di eccitazione muscolare, necessario per la successiva normalizzazione degli inviluppi lineari, è stato ricercato tra tutti i trial di camminata a disposizione.

3.2 Metodi

Nella seguente sezione rappresentante i metodi, verrà descritto dettagliatamente il workflow utilizzato e le simulazioni di tracciamento e predittive eseguite.

3.2.1 Workflow

Il workflow messo a punto per la generazione di simulazioni predittive si basa interamente su OpenSim Moco (Figura 9) e consta di due step fondamentali:

1. definizione della condizione iniziale (*MocoTrack*),
2. realizzazione della simulazione predittiva, a partire dalla soluzione di tentativo iniziale.

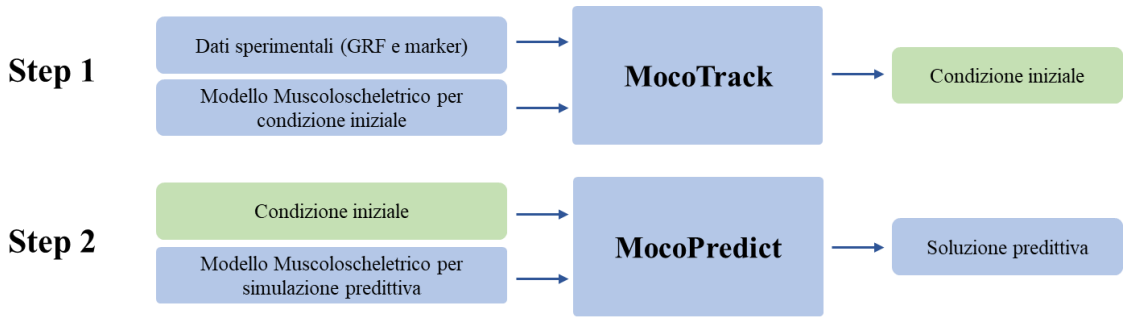

Figura 9: Workflow per le simulazioni predittive. Costituito da due step: il primo consiste nella realizzazione di una condizione iniziale, utilizzabile successivamente nel secondo step come soluzione di primo tentativo per le simulazioni predittive sucessive.

Tutte le operazioni eseguite all'interno di OpenSim Moco sono state condotte a partire da codici (script) custom made, che sfruttano l'interfaccia di programmazione del software per Matlab (OpenSim Moco API).

3.2.2 Modelli muscoloscheletrici usati

Per entrambi gli step del workflow sono state utilizzate delle versioni scalate del modello Rajagopal2015 e successivamente modificate. Rispetto al modello di riferimento (Full Body Model) sono state apportate diverse modifiche al fine di semplificare il modello in vista delle simulazioni predittive. In primo luogo, è stato ridotto il numero di corpi (ossa), e muscoli, per ridurre i tempi di calcolo, pur mantenendo almeno un muscolo per gruppo muscolare (in base al ruolo anatomo/fisiologico). Per garantire la riuscita della simulazione mediante collocazione diretta i muscoli sono sostituiti passando da modello di Millard 2012 [43] a quello di De Groote Fregly 2016 [34]. Dopodiché sono stati bloccati alcuni giunti sottoattuati o non di immediata rilevanza (i.e. articolazione metatarsale (mtp) e subtalare (subtalar)) il cui inserimento avrebbe potuto aumentare la complessità generale e i tempi computazionali. Infine, al modello relativo alla KGC6, sono state inserite delle sfere di contatto (*SmoothSphereHalfSpaceForce*) [44] in corrispondenza del calcagno e dell'avampiede. Quest'ultima modifica si è resa necessaria, poiché non essendo fornito in ingresso una forza di contatto, senza l'inserimento di questi elementi, il modello sarebbe sprofondato nello spazio (per l'azione della forza di gravità).

Tabella 1: Parametri degli elementi sferici introdotti per simulare il contatto.

Parametri sfere di contatto	
stiffness	3067776 N/m^2
dissipation	2 s/m
static_friction	0.8
dynamic_friction	0.8
viscous_friction	0.8
transition_velocity	0.2 m/s
derivative_smoothing	1e-5
herz_smoothing	300
hunt_crossley_smoothing	50

In OpenSim Moco gli stati e controlli utilizzati nella funzione costo sono relativi alle variabili del modello, cioè rappresentano le coordinate generalizzate $q(t)$ (angoli articolari), velocità generalizzate $u(t)$, attivazioni $a(t)$ ed eccitazioni $e(t)$.

Il modello per la condizione iniziale è costituito da 64 stati (21 coordinate generalizzate $q(t)$ corrispondenti a ciascuna articolazione e ai 3 gradi di libertà del bacino, 21 velocità generalizzate $u(t)$ e da 22 attivazioni muscolari $a(t)$) e da 41 controlli (eccitazioni $e(t)$: 3 per gli attuatori lombari, 22 per i muscoli e 16 per gli attuatori di riserva posti nelle articolazioni) , mentre il modello per le simulazioni predittive presenta 25 controlli (eccitazioni $e(t)$: 3 per gli attuatori lombari, 22 per i muscoli), a cui si aggiungono due vincoli cinematici relativi al moto della patella.

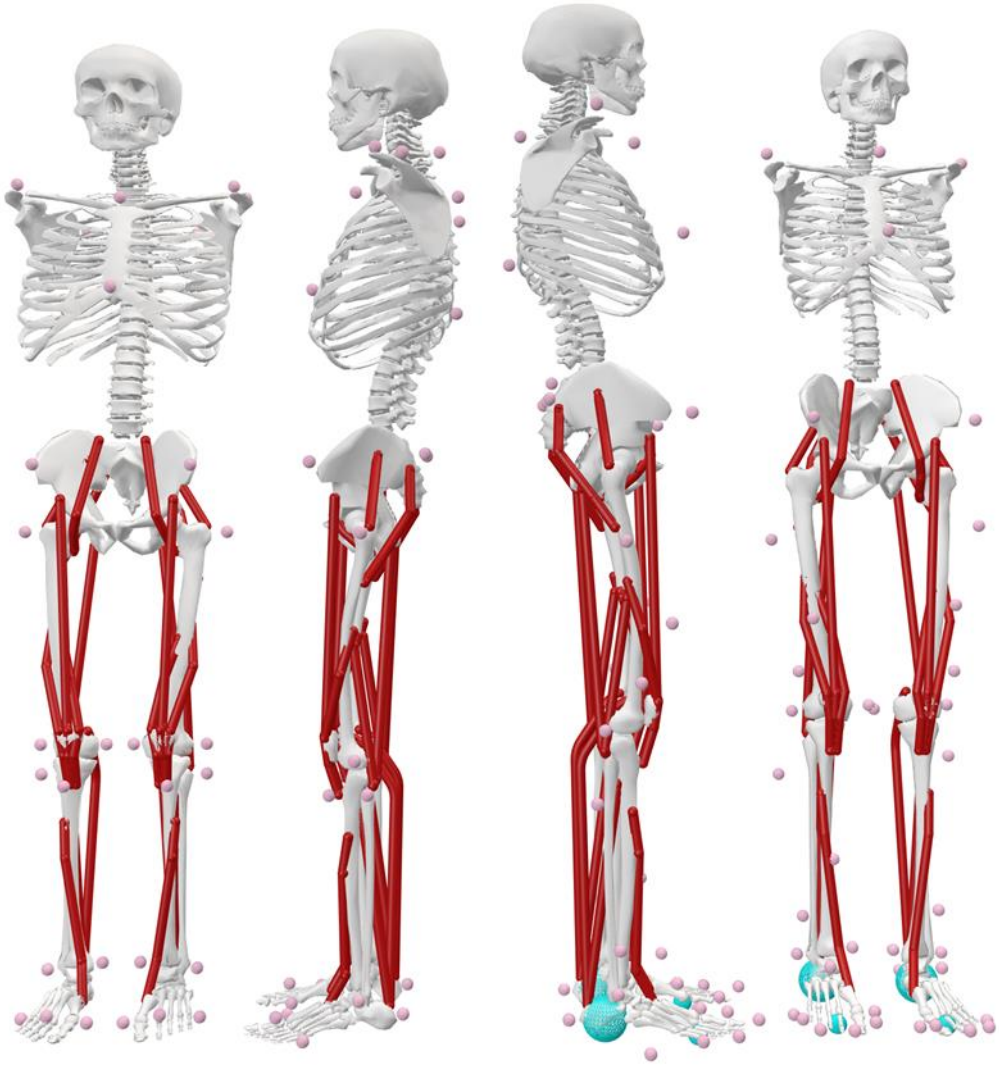

Figura 10: Modello Rajagopal2015 scalato e modificato. A sinistra è mostrato il modello relativo al soggetto 2014001 del dataset online, a destra quello relativo al soggetto della KGC6. I corpi sono stati ridotti a 14 rimuovendo: omero, ulna, radio e mano. I muscoli sono stati ridotti a 11 per gamba, i rimanenti sono: gluteo massimo (2), gluteo medio (2), psoas, bicipite femorale lungo e breve, retto femorale, vaso mediale e laterale, gastrocnemio mediale, soleo e tibiale anteriore. Il modello della KGC6 presenta sfere di contatto poste sul calcagno e nell'avampiede.
Essendo il Full Body Model composto da muscoli, come gluteo massimo e medio, aventi più capi, il termine (2) indica il numero dell'unità considerata.

3.2.3 Step 1: Definizione della condizione iniziale

La realizzazione di una condizione iniziale è necessaria per ottenere la convergenza dell'algoritmo di risoluzione e l'uso di dati sperimentali relativi a soggetti sani è fondamentale per la generazione di una condizione iniziale affidabile da utilizzare per le simulazioni predittive successive.
Per definire la condizione iniziale per la simulazione predittiva ci si è avvalsi del tool *MocoTrack*. In questo modo, si è ottenuta una condizione iniziale in linea con un set di dati sperimentali, quindi rappresentativa della cinematica e dell'attivazione muscolare proprie di un soggetto sano che cammina a velocità 'normale' (0.8 - 1.2 ms^{-1}). Questa soluzione è stata successivamente utilizzata come soluzione di primo tentativo per il problema predittivo. Inoltre, essendo la soluzione generata da un tool di Moco, il file di output ottenuto si presenta nel formato richiesto per le simulazioni successive, semplificandone l'utilizzo.
Di seguito sono descritte in dettaglio le procedure messe in atto per eseguire questa prima simulazione in *MocoTrack*.

Come descritto nel capitolo precedente (sezione 2.3), l'esecuzione di una simulazione in Moco richiede la definizione del problema da risolvere, sotto forma di una funzione costo composta di più termini (goals). In particolare, al fine (i) di garantire il tracciamento dei marker, (ii) la minimizzazione dei controlli e (iii) di vincolare allo stesso valore iniziale le attivazioni ed eccitazioni, sono stati utilizzati tre *Goals*:

- *MocoMarkerTrackingGoal*,
- *MocoControlGoal*,
- *MocoInitialActivationGoal*.

La funzione costo risultante è riportata in Eq.3.1 ed è data dalla somma della funzione costo J_1 (*MocoMarkerTrackingGoal*) e J_2 (*MocoControlGoal*):

$$J = J_1 + J_2 = w_{marker} \sum_{k=1}^{52} \left(m_{S_k} - w_k \cdot m_{M_k}\right)^2 + w_{effort} \frac{\int_{t_i}^{t_f} \sum_{i=1}^{41} w_i \cdot |x_i(t)|^p \, dt}{q_{pelvis}(t_f) - q_{pelvis}(t_i)} \quad (3.1)$$

dove:

- w_{marker} : peso della funzione *MocoMarkerTrackingGoal*
- w_{effort}: peso della funzione *MocoControlGoal*
- w_k: peso del marker k-esimo del modello
- w_i: peso del controllo i-esimo
- m_{S_k}: posizione del marker k-esimo sperimentale
- m_{M_k}: posizione del marker k-esimo del modello
- $x_i(t)$: controllo i-esimo corrispondente all'eccitazione $e_i(t)$
- p: esponente, posto pari a 2
- $q_{pelvis}(t_f) - q_{pelvis}(t_i)$: differenza tra la posizione finale e iniziale della pelvi
- t_i: tempo iniziale
- t_f: tempo finale

Il problema di tracciamento è stato risolto riducendo al minimo la funzione J soggetta a vincoli che impongono la cinematica scheletrica, la dinamica scheletrica e la dinamica di attivazione muscolare:

$$u(t) = \dot{q}(t) \quad (3.2)$$

$$\dot{u}(t) = f\big(q(t), u(t), a(t)\big) \quad (3.3)$$

$$\dot{a}(t) = f\big(e(t), a(t)\big) \quad (3.4)$$

I limiti sono stati posti sugli stati e sui controlli. Per gli angoli articolari sono stati mantenuti i limiti superiore e inferiore specificati nel modello muscoloscheletrico, mentre le velocità generalizzate $u(t)$ sono state limitate tra -50 rad/s e 50 rad/s, e le eccitazioni (controlli) $e(t)$ e le attivazioni $a(t)$ tra 0.01 e 1.

Il peso della funzione *MocoControlGoal* è stato lasciato a 0.001 e a 1 (valori di default) per i singoli controlli, mentre il peso della funzione *MocoMarkerTrackingGoal* è stato impostato pari a 20 così da garantire un corretto tracciamento dei file sperimentali. Infine, per i singoli marcatori è stato scelto un peso tale da dare maggior importanza ai marker più affidabili (ad es. i marker posizionati sulla cresta iliaca anteriore e posteriore), ossia i marker più facilmente individuabili per palpazione (Tabella 2).

Tabella 2: Peso di tracciamento applicato ai marker del modello.

Marker	Weight	Marker	Weight
R_SAA	100	*R_FLE*	20
L_SAA	100	*R_FME*	20
R_SAE	100	*R_FCC*	10
L_SAE	100	*R_FM1*	5
CV7	3	*R_FM5*	5
SJN	3	*L_FTC*	0
TV10	1	*L_FLE*	20
R_IAS	100	*L_FME*	20
L_IAS	100	*L_FCC*	10
R_IPS	100	*L_FM5*	5
L_IPS	100	*R_FM2*	5
R_FTC	0	*L_FM2*	5

I valori obiettivo complessivi dei goal presenti sono riportati nella tabella seguente:

Tabella 3: Valore obiettivo delle funzioni costo usate nel MocoTrack.

Funzione obiettivo	Obiettivo
MocoMarkerTrackingGoal J_1	8.865239
MocoControlGoal J_2	0.379781
Funzione costo complessiva J	9.245020

Il solutore usato è quello impostato di default per il *MocoTrack*, cioè *MocoCasADiSolver*, le cui impostazioni sono riportate in Tabella 4.

Tabella 4: Impostazioni del solutore per il MocoTrack.

Impostazioni MocoCasADiSolver	Valori
multibody_dynamics_mode	explicit
transcription_scheme	Hermite-Simpson
optim_convergence_tolerance	1e-2
optim_constraint_tolerance	1e-2
optim_sparsity_detection	random
optim_finite_difference_scheme	‘forward’

Per completezza, in appendice è riportato lo script commentato, così da chiarire come la trattazione matematica qui riportata è tradotta in forma di codice Matlab.

3.2.4 Step2: Simulazione predittiva

Realizzata la condizione iniziale è stato possibile creare il *MocoStudy* personalizzato per le simulazioni predittive.
L'obiettivo finale era quello di ottenere una camminata quanto più normale possibile (cioè non contraddistinta da movimenti innaturali quali eccessive rotazioni degli arti o camminate 'saltellanti'). A tal fine si è proceduto per passi, generando in totale quattro diverse simulazioni predittive, la cui funzione costo è stata via via complicata attraverso l'aggiunta di nuovi Moco *Goals*.

Le simulazioni predittive sono state realizzate personalizzando il *MocoStudy*, con l'uso di un modello muscoloscheletrico (scalato sul soggetto della KGC6) e della condizione iniziale precedentemente generata tramite il *MocoTrack*.
Sono state eseguite quattro simulazioni con aggiunta progressiva di funzioni costo differenti in modo tale da aumentare la complessità del problema di optimal control, al fine di predire il task motorio desiderato (camminata 'normale').

La prima simulazione presenta un'unica funzione costo, *MocoAverageSpeedGoal*, che rappresenta un vincolo di endpoint in cui la velocità media finale del soggetto deve essere pari al valore impostato (1.0 ms^{-1}).
Nella seconda simulazione a *MocoAverageSpeedGoal* è stato aggiunto un vincolo di periodicità (*MocoPeriodicityGoal*), che forza i valori iniziali e finali delle variabili in gioco ad essere equivalenti. Questo vincolo ha lo scopo di garantire la periodicità nel passo.
La terza simulazione presenta un'ulteriore funzione costo, *MocoControlGoal* (J_1), per minimizzare i controlli (o eccitazioni), in aggiunta ai vincoli di endpoint precedenti.
La quarta simulazione è implementata aggiungendo ai goals precedenti la funzione costo *MocoSumSquaredStateGoal* (J_2) per minimizzare le attivazioni muscolari.

Nell'Eq.3.5 è mostrata la funzione compressiva, distinguendo i vari goal usati nelle simulazioni (la terza costituita dalla sola J_1 e la quarta dalla somma di J_1 e J_2).

$$J = J_1 + J_2 = w_{effort} \frac{\int_{t_i}^{t_f} \sum_{i=1}^{41} w_i \cdot |x_i(t)|^p \, dt}{q_{pelvis}(t_f) - q_{pelvis}(t_i)} + w_{activation} \int_{t_i}^{t_f} \sum_{s \in S} w_s y_s(t)^2 dt \quad (3.5)$$

dove:

- w_{effort}: peso della funzione *MocoControlGoal*
- $w_{activation}$: peso della funzione obiettivo *MocoSumSquaredStateGoal*
- S: set di variabili di stato usate nell'obiettivo
- w_s: peso per la variabile di stato *s*
- $y_s(t)$: variabile di stato
- w_i: peso del controllo i-esimo
- $x_i(t)$: controllo i-esimo corrispondente all'eccitazione $e_i(t)$
- p: esponente, posto pari a 2
- $q_{pelvis}(t_f) - q_{pelvis}(t_i)$: differenza tra la posizione finale e iniziale della pelvi
- t_i: tempo iniziale
- t_f: tempo finale

Il problema di predizione, per le prime due simulazioni, è stato risolto rispettando i vincoli che impongono la cinematica scheletrica, dinamica scheletrica e dinamica di attivazione muscolare, mentre per le ultime due simulazioni in aggiunta avviene la minimizzazione della funzione costo J in rispetto dei vincoli sopraelencati:

$$u(t) = \dot{q}(t) \quad (3.6)$$

$$\dot{u}(t) = f\big(q(t), u(t), a(t)\big) \quad (3.7)$$

$$\dot{a}(t) = f\big(e(t), a(t)\big) \quad (3.8)$$

I limiti sono stati posti sugli stati e sui controlli. Per gli angoli articolari sono stati mantenuti i limiti superiore e inferiore specificati nel modello muscoloscheletrico, mentre le velocità generalizzate $u(t)$ sono state limitate tra -50 rad/s e 50 rad/s, e le eccitazioni (controlli) $e(t)$ e le attivazioni $a(t)$ tra 0.01 e 1.

Il peso della funzione *MocoControlGoal* è stato impostato pari a 10, mentre per la funzione *MocoSumSquaredStateGoal* pari a 5. I pesi relativi ai singoli controlli e stati minimizzati dai rispettivi *Goals* non sono stati personalizzati, ma lasciati pari a 1.

In Tabella 5 vengono riportati i valori limite per gli angoli articolari.

Tabella 5: Limiti articolari inferiori e superiori imposti nel MocoPredict.

Angoli articolari	Limiti inferiori e superiori
pelvis_Tilt – pelvis_List – pelvis_Rotation	[-5°;5°]
pelvis_tx	[0m;1.5m]
pelvis_ty	[0.94m;1m]
pelvis_tz	[0m;0.5m]
hip_flexion_r - hip_flexion_l	[-10°;35°]
hip_adduction_r - hip_adduction_l	[-10°;10°]
hip_Rotation_r - hip_Rotation_l	[-15°;0°]
Lumbar_extension	[-25°; -15°]
Lumbar_bending	[-6°; 6°]
Lumbar_Rotation	[-10°; 10°]
knee_angle_r - knee_angle_l	[0°; 65°]
ankle_angle_r - ankle_angle_l	[-10°; 25°]

Gli obiettivi da raggiungere per le varie simulazioni sono riportati in Tabella 6.

Tabella 6: Valore obiettivo delle funzioni costo usate nelle varie simulazioni predittive.

Funzione obiettivo	Obiettivo
Simulazione 1	
MocoAverageSpeedGoal	0
Simulazione 2	
MocoAverageSpeedGoal	0
MocoPeriodicityGoal	0
Simulazione 3	
MocoAverageSpeedGoal	0
MocoPeriodicityGoal	0
MocoControlGoal	32.424795
Simulazione 4	
MocoAverageSpeedGoal	0
MoxoPeriodicityGoal	0
MocoControlGoal	34.021015
MocoSumSquaredStateGoal	7.566730
Funzione costo complessiva	41.587744

Anche in questo caso si è scelto *MocoCasADiSolver* come solutore per la risoluzione del problema di optimal control, impostando i parametri come indicato in Tabella 7.

Tabella 7: Impostazioni del solutore del MocoPredict.

Impostazioni MocoCasADiSolver	Valori
multibody_dynamics_mode	explicit
transcription_scheme	Hermite-Simpson
optim_convergence_tolerance	1e-1
optim_constraint_tolerance	1e-1
optim_sparsity_detection	random
optim_finite_difference_scheme	'forward'

3.2.5 Ulteriori analisi svolte sui tools di OpenSim Moco

Essendo il progetto di tesi principalmente di carattere esplorativo, data la novità del software OpenSim Moco nel panorama delle simulazioni predittive, prima della realizzazione della condizione iniziale sono state svolte ulteriori analisi sui tools *MocoTrack* e *MocoInverse*, così da comprenderne meglio le basi ed il funzionamento.

In entrambi i casi sono stati utilizzati i dati relativi alla sesta edizione della "Grand Challenge Competition", impiegati per le simulazioni predittive. Parimenti, il modello muscoloscheletrico utilizzato corrisponde ad una versione scalata (alle dimensioni del soggetto sotto esame) del modello generico Full Body Model.

MocoInverse

L'analisi del tool *MocoInverse* si è concentrata su due aspetti fondamentali: la risoluzione di un problema inverso standard per generare le attivazioni muscolari partendo da una cinematica nota e l'implementazione del punto precedente con un obiettivo di tracciamento per i dati elettromiografici. Lo scopo di queste analisi era, rispettivamente, valutare il grado di precisione nella generazione delle attivazioni muscolari rispetto a dati elettromiografici e valutare l'effetto di pesi differenti nella funzione costo per il tracciamento di dati elettromiografici.

La prima simulazione presenta il tool base costituito da due funzioni obiettivo principali: *MocoControlGoal* (J_1) e *MocoInitialActivationGoal*, per vincolare l'attivazione iniziale

in modo che sia uguale all'eccitazione iniziale (evita picchi di attivazione iniziale ed è un vincolo di endpoint).
La seconda simulazione presenta l'aggiunta di *MocoControlTrackingGoal* (J_2) per tracciare i dati sperimentali di elettromiografia.

Nell'Eq.3.13 è mostrata la funzione compressiva, distinguendo i vari goal usati nelle simulazioni (la prima costituita dalla sola J_1 e la seconda dalla somma di J_1 e J_2).

$$J = J_1 + J_2 =$$

$$= w_{controlTrack} \int_{t_i}^{t_f} \sum_{i=1}^{41} w_i \cdot \left\| x_{S_i} - x_{M_i} \right\|^2 + w_{effort} \int_{t_i}^{t_f} \sum_{i=1}^{41} w_i \cdot |x_i(t)|^p \, dt \quad (3.13)$$

dove:

- w_{effort}: peso della funzione *MocoControlGoal*
- $w_{controlTrack}$: peso della funzione *MocoControlTrackingGoal*
- w_i: peso del controllo i-esimo
- x_{S_i}: controllo i-esimo sperimentale
- x_{M_i}: controllo i-esimo del modello
- $x_i(t)$: controllo i-esimo corrispondente all'eccitazione $e_i(t)$
- p: esponente, posto pari a 2
- t_i: istante di tempo iniziale
- t_f: istante di tempo finale

Entrambi i problemi inversi sono risolti riducendo al minimo la funzione costo soggetta a vincoli che impongono la cinematica scheletrica, dinamica scheletrica e dinamica di attivazione muscolare. I limiti sono stati posti sugli stati e controlli, gli angoli articolari presentano limite superiore e inferiore specificato nel modello muscoloscheletrico, le velocità generalizzate $u(t)$ sono comprese tra -50 rad/s e 50 red/s, le eccitazioni (controlli) $e(t)$ e le attivazioni $a(t)$ sono comprese tra 0.01 e 1.

Il peso usato per *MocoControlGoal* in entrambe le simulazioni è di 1 ed ogni singolo controllo presenta lo stesso peso di 1; nella seconda simulazione sono stati invece testati pesi differenti (1, 5, 10, 25, 50 e 100) per l'obiettivo *MocoControlTrackingGoal*.

Il solutore usato è quello impostato di default per il *MocoInverse* (*MocoCasADiSolver*), i cui parametri sono riportati in Tabella 8.

Tabella 8: Impostazioni del solutore del MocoInverse.

Impostazioni MocoCasADiSolver	Valori
multibody_dynamics_mode	implicit
interpolate_control_midpoints	false
minimize_implicit_auxiliary_derivatives	true
implicit_auxiliary_derivatives_weight	0.01
optim_convergence_tolerance	1e-3
optim_constraint_tolerance	1e-3
optim_sparsity_detection	random
optim_finite_difference_scheme	'forward'

MocoTrack

L' analisi del tool *MocoTrack* si è concentrata su due aspetti fondamentali. Il primo è il tracciamento di dati sperimentali con infittimento progressivo della mesh temporale; il secondo è il tracciamento di dati sperimentali con l'aggiunta di segnali elettromiografici. Lo scopo di queste analisi è valutare gli effetti dell'infittimento della mesh temporale in termini cinematici e valutare il grado precisione, in termini cinematici e di attivazione, ottenuto nel tracciamento assistito da dati elettromiografici.

La prima simulazione presenta il tool base costituito da due funzioni obiettivo principali: *MocoMarkerTrackingGoal* e *MocoControlGoal*. La funzione costo complessiva è data dall'unione della funzione costo J_1 (*MocoMarkerTrackingGoal*) e J_2 (*MocoControlGoal*).

Nella seconda simulazione, oltre al tool base, è stato aggiunto l'obiettivo *MocoControlTrackingGoal* per tracciare i dati sperimentali di elettromiografia.

Nell'Eq.3.14 è mostrata la funzione compressiva, distinguendo i vari goal usati nelle simulazioni (la prima costituita dalla somma di J_1 e J_2, la seconda dalla somma di J_1, J_2 e J_3).

$$J = J_1 + J_2 + J_3 =$$

$$= w_{track} \sum_{k=1}^{52} \left(m_{S_k} - w_k \cdot m_{M_k}\right)^2 + w_{effort} \frac{\int_{t_i}^{t_f} \sum_{i=1}^{41} w_i \cdot |x_i(t)|^p \, dt}{q_{pelvis}(t_f) - q_{pelvis}(t_i)}$$

$$+ w_{controlTrack} \int_{t_i}^{t_f} \sum_{i=1}^{41} w_i \cdot \left\| x_{S_i} - x_{M_i} \right\|^2 \quad (3.14)$$

dove:

- w_{track}: peso della funzione *MocoMarkerTrackingGoal*
- w_{effort}: peso della funzione *MocoControlGoal*
- $w_{controlTrack}$: peso della funzione *MocoControlTrackingGoal*
- w_k: peso del marker k-esimo del modello
- w_i: peso del controllo i-esimo
- m_{S_k}: posizione del marker k-esimo sperimentale
- m_{M_k}: posizione del marker k-esimo del modello
- x_{S_i}: controllo i-esimo sperimentale
- x_{M_i}: controllo i-esimo del modello
- $x_i(t)$: controllo i-esimo corrispondente all'eccitazione $e_i(t)$
- p: esponente, posto pari a 2
- $q_{pelvis}(t_f) - q_{pelvis}(t_i)$: differenza tra la posizione finale e iniziale della pelvi
- t_i: istante di tempo iniziale
- t_f: istante di tempo finale

Il problema di tracciamento è risolto riducendo al minimo la funzione costo soggetta a vincoli che impongono la cinematica scheletrica, dinamica scheletrica e dinamica di attivazione muscolare; limiti sono stati posti sugli stati e controlli, gli angoli articolari presentano limite superiore e inferiore specificato nel modello muscoloscheletrico, le velocità generalizzate $u(t)$ sono comprese tra -50 rad/s e 50 red/s, le eccitazioni (controlli) $e(t)$ e le attivazioni $a(t)$ sono comprese tra 0.01 e 1.

Il peso generale del *MocoTrack* è stato impostato pari a 10, così da garantire un corretto tracciamento dei file sperimentali; nella funzione *MocoControlGoal* viene lasciato il peso preimpostato di 0.001 per la funzione e 1 per i singoli controlli; per la funzione *MocoMarkerTrackingGoal* il peso dei singoli marker è riportato nella tabella seguente.

Tabella 9: Peso di tracciamento applicato ai marker del modello

Marker	Weight	Marker	Weight
Sternum	100	*RThighSuperior*	5
Neck	100	*RThighInferior*	25
RShoulder	100	*RPatella*	100
LShoulder	100	*RShankSuperior*	10
Xiphoid	50	*RShankInferior*	10
Thoracic	100	*RHell*	60
Lumbar	100	*RMidfootMedial*	25
Sacral	100	*RMidfootLateral*	25
RAsis	50	*RMidfootSuperior*	25
LAsis	50	*RHindfoot*	25
RPsis	100	*RToeMedial*	25
LPsis	100	*RToeLateral*	25
RToe	30	*LMidfootLateral*	20
LThighSuperior	12	*LMidfootSuperior*	20
LThighInferior	10	*LHindfoot*	20
LThighLateral	10	*LToeMedial*	20
LPatella	100	*LToeLateral*	20
LShankSuperior	10	*LToe*	2
LShankInferior	10	*LShanLateral*	10
LHell	50	*LMidfootMedial*	20

Per la simulazione con dati elettromiografici il peso della funzione *MocoControlTrackingGoal* viene imposto pari a 100.

Il solutore usato è quello impostato di default per il *MocoTrack* (*MocoCasADiSolver*), nella tabella seguente vengono riportati i parametri del solutore:

Tabella 10: Impostazioni del solutore del MocoTrack.

Impostazioni MocoCasADiSolver	Valori
multibody_dynamics_mode	explicit
transcription_scheme	Hermite-Simpson
optim_convergence_tolerance	1e-2
optim_constraint_tolerance	1e-2
optim_sparsity_detection	random
optim_finite_difference_scheme	‘forward’

3.2.6 Analisi dei dati

I risultati ottenuti sono stati in un primo momento valutati da un punto di vista qualitativo, osservando la camminata riprodotta, alla ricerca di evidenti errori macroscopici (es. camminata caratterizzata da eccessiva flessione delle ginocchia, strisciamento, balzelli, etc). In un secondo momento, è stata condotta un'analisi quantitativa al fine di ottenere una valutazione più puntuale che permettesse di evidenziare criticità specifiche (o discrepanza rispetto a valori noti o considerati attendibili). A tale scopo sono stati calcolati due indici statistici utilizzati per valutare la bontà delle simulazioni predittive condotte (in confronto ai dati sperimentali), e per analizzare i vari tool di OpenSim Moco: il coefficiente di determinazione (R^2) e l'errore quadratico medio (o Root Mean Squared Error, RMSE).

Il primo (R^2) valuta quanta differenza è presente tra i valori osservati di y nel campione ed i valori che il modello ha stimato per y, con valore minimo pari a 0 e valore massimo (perfetta corrispondenza) pari a 1. L'indice R^2 permette quindi di quantificare la similarità dei profili analizzati (es. inviluppo lineare del segnale EMG ed eccitazione muscolare predetta da Moco). In altre parole, ci dice quanto la forma delle curve sia simile (se entrambe crescono in modo simile - i.e. stessa pendenza - allo stesso tempo R^2 sarà prossimo a 1, altrimenti sarà più prossimo a 0). Più è grande il valore di R^2, più il modello ha un alto potere predittivo, cioè il modello con il coefficiente di determinazione maggiore sarà quello che avrà minori discrepanze tra i valori osservati e quelli attesi della y.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2} \quad (3.15)$$

dove: y_i: sono i dati osservati, $\bar{y}_i$: è la media dei dati osservati e $\bar{y}_i$: sono i dati stimati dal modello.

Il secondo (RMSE) indica la discrepanza quadratica media fra i valori dei dati osservati ed i valori dei dati stimati ed è calcolato elevando al quadrato la differenza tra il valore reale y e quello predetto.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}} \quad (3.16)$$

dove: y_i: sono i dati osservati, $\overline{y_i}$: sono i dati stimati dal modello e n: numero di dati.

Gli indici sopra descritti sono stati usati per valutare la cinematica e le attivazioni generate da Moco a confronto con i risultati di Inverse Kinematics di OpenSim e i dati di elettromiografia, precedentemente elaborati tramite MatLab per generare un valore medio e deviazione standard sui vari trial di camminata a disposizione (n = 6) interpolati su 101 punti.

3.2.7 Macchina usata per le simulazioni

Per le simulazioni presenti nel workflow è stato utilizzato un computer Windows, presentante processore AMD Ryzen™ 5 5600X (3.7GHz, con 6 core CPU e 12 threads) e Ram Crucial Ballistix DDR4 4x8Gb 3600MHz.

Capitolo 4: Analisi risultati

Nel seguente capitolo verranno esposti e analizzati i risultati ottenuti dalle simulazioni di tracciamento (Step 1, Workflow, Figura 9) e predizione (Step 2).
La valutazione dei risultati generati da Moco è eseguita tramite un confronto (i) tra la cinematica predetta (da *MocoPredict*) e la cinematica ottenuta tramite un'analisi di Inverse Kinematics svolta con OpenSim, e (ii) tra le eccitazioni muscolari predette ed i dati sperimentali di elettromiografia.
In primo luogo, sarà esposto un confronto qualitativo della camminata, per poi passare ad analizzare graficamente i risultati ottenuti e infine, tramite l'uso degli indici introdotti nel capitolo precedente, sarà possibile dare un'interpretazione matematica all'analisi grafica.

4.1 Step1: Analisi risultati della condizione iniziale

Come anticipato nel Capitolo 2, il tool *MocoTrack* fornisce come output i dati relativi alla cinematica e alle attivazioni muscolari necessarie per svolgere il task motorio rappresentativo dei dati sperimentali tracciati. Essendo la condizione iniziale fondamentale per la riuscita della simulazione predittiva è necessario analizzare le quantità d'interesse tracciate.

4.1.1 Analisi qualitativa della camminata tracciata

Di seguito sono riportati sette frame della camminata simulata, a percentuali del ciclo del passo differenti (0%, 0% - 50%, 50%, 50% - 100% e 100%), con *MocoTrack* (costituito dai seguenti *Goals* *MocoMarkerTrackingGoal*, *MocoControlGoal* e *MocoInitialActivation*) per valutare visivamente la corrispondenza ottenuta tra i marker sperimentali e quelli del modello.

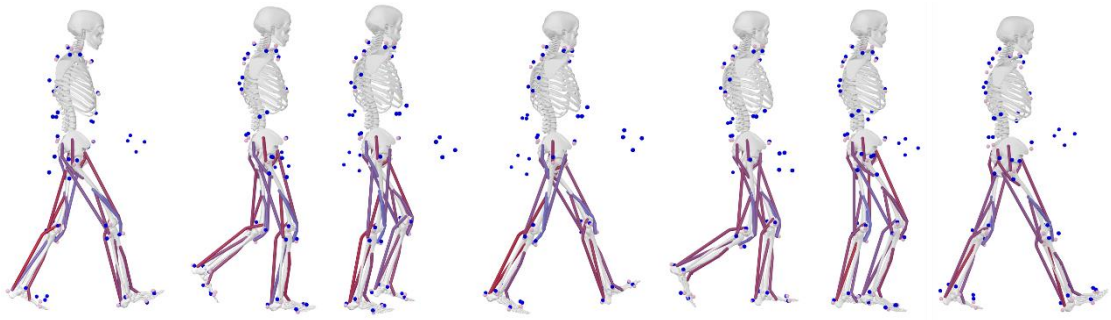

Figura 11: Frame relativi alla camminata tracciata da MocoTrack. Le varie percentuali del ciclo del passo (es. 0%, 0-50%, 50%, 50-100% e 100%), tracciato da Moco con il modello muscoloscheletrico scalato sul soggetto, sono rappresentate attraverso frame.

Visivamente la camminata tracciata da Moco risulta normale rispetto al movimento svolto dai marker sperimentali.

4.1.2 Analisi grafica e statistica dei risultati tracciati

Gli angoli lombari di rotazione e di flessione laterale (bending) non presentano sostanziali differenze rispetto ai valori di Inverse Kinematics, al contrario l'angolo di flesso-estensione lombare presenta un andamento differente soprattutto nella prima fase del ciclo del passo, cioè dallo 0% al 40%. Tuttavia, l'escursione risulta ridotta (circa 4°) e l'errore assoluto risulta relativamente ridotto, nell'ordine di 2°.

Gli angoli articolari di flessione e adduzione dell'anca sono tracciati coerentemente con OpenSim, mentre si nota una differenza evidente per quel che riguarda l'angolo di intra-extrarotazione del femore rispetto al bacino. Al 60% del ciclo del passo, nella simulazione predittiva la gamba destra presenta un'extra rotazione inferiore (cieca 7°) rispetto ai valori stimati dalla Inverse Kinematics. Similmente, la gamba sinistra inizia con valori di extra rotazione inferiori (circa 6° in meno) rispetto ai valori di OpenSim.

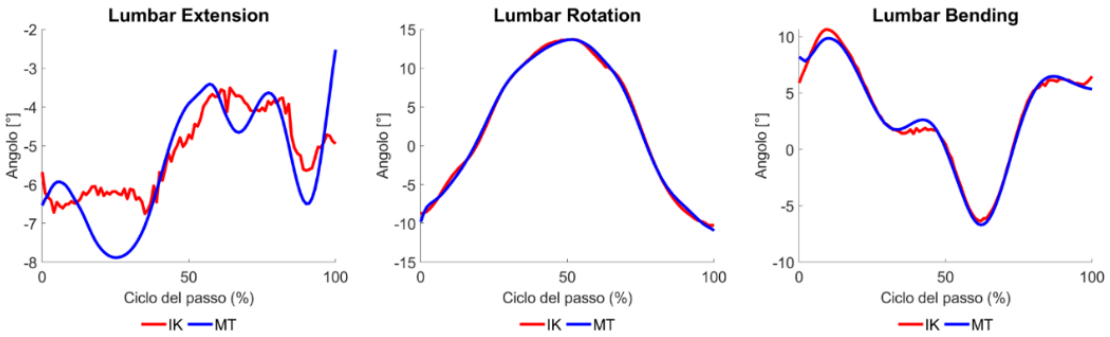

Figura 12: Confronto tra gli angoli articolari lombari di estensione, rotazione ed estensione laterale (bending) ottenuti dall'Inverse Kinematics (dati Dataset Online) di OpenSim (rosso) e tracciati da Moco (blu).

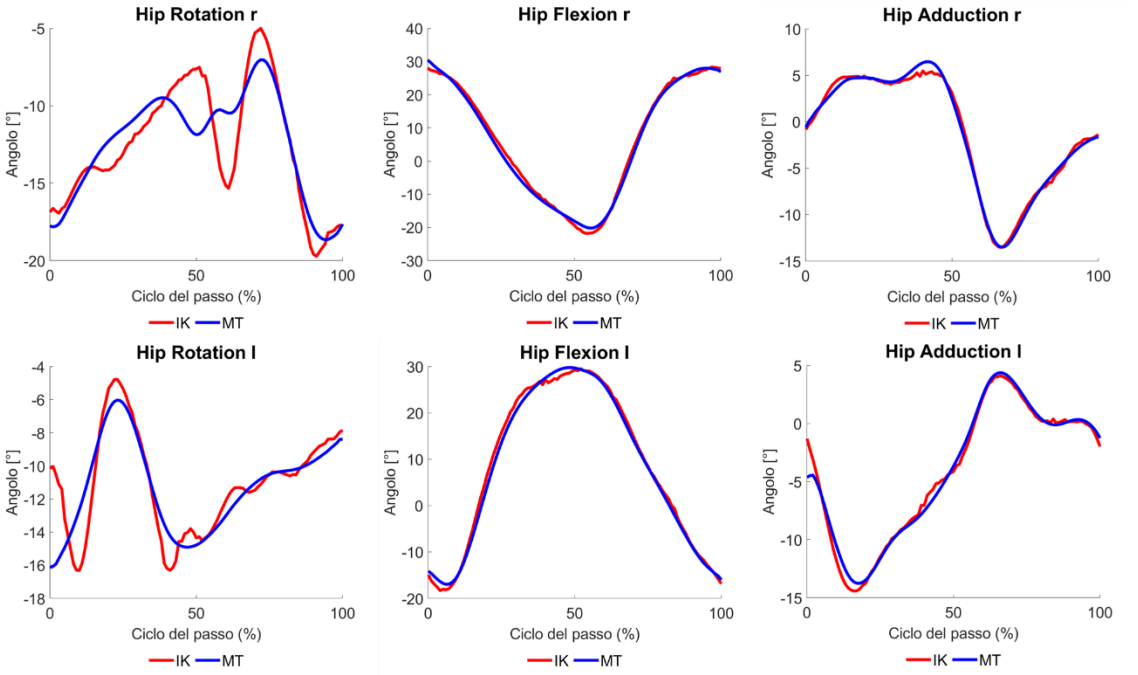

Figura 13: Confronto tra gli angoli articolari di rotazione, flessione e adduzione dell'anca ottenuti con OpenSim (rosso) e Moco (blu).

L'angolo di flesso-estensione del ginocchio (Figura 14) è stato tracciato da Moco con valori e curve quasi identici all'Inverse Kinematics di OpenSim, seppur si notano differenze minime durante la fase finale del supporto monopodalico (attorno al 50% del ciclo del passo).

L'angolo di flessione dorsale della caviglia (Figura 15) tracciato dal *MocoTrack* presenta differenza in termini di picchi massimi (flessione dorsale) e minimi (flessione plantare) rispetto ai valori di OpenSim. Tra lo 0% e 20% del ciclo del passo per la caviglia sinistra presenta una differenza di circa 15° e nell'intorno del 50% di 5°, tra il tracciamento di Moco e la simulazione di cinematica inversa; al contrario, la caviglia destra presenta una differenza di circa 10°, tra il 50% e 60% del ciclo del passo.

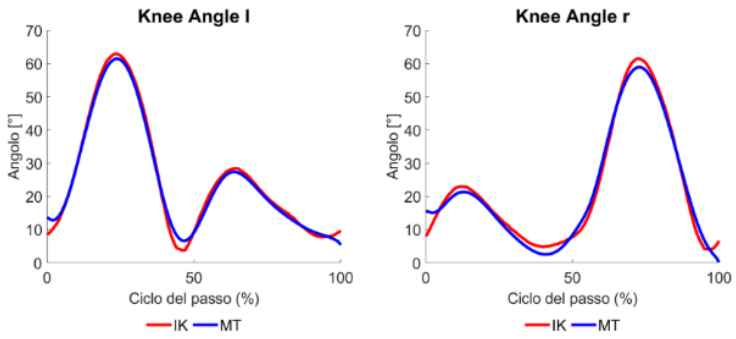

Figura 14: Confronto tra gli angoli di flesso-estensione del ginocchio ottenuti dall'Inverse Kinematics (dati Dataset Online) di OpenSim (rosso) e tracciati da Moco (blu).

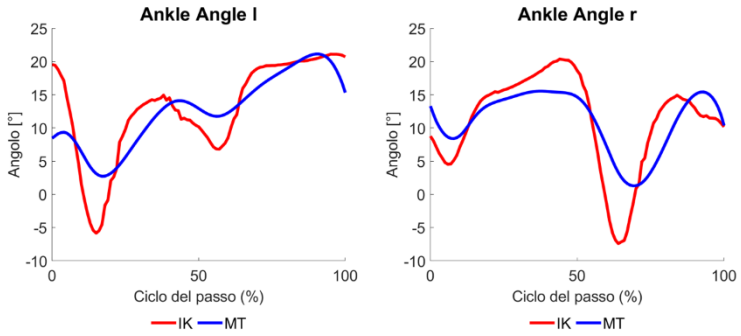

Figura 15: Confronto tra gli angoli articolari di flesso-estensione della caviglia ottenuti dall'Inverse Kinematics (dati Dataset Online) di OpenSim (rosso) e tracciati da Moco (blu).

Dai grafici si evince che gli angoli tracciati da Moco seguono ragionevolmente bene i valori ottenuti tramite l'Inverse Kinematics (considerati valori sperimentali), tuttavia vi sono alcune differenze non del tutto trascurabili a livello dell'angolo di intra/extrarotazione dell'anca, di flesso-estensione della caviglia e di estensione lombare. Per avere un confronto numerico, in Tabella 11, sono riportati i valori di R^2 e RMSE calcolati.

Tabella 11: Valori di R^2 e RMSE per la cinematica ottenuta da MocoTrack a confronto con OpenSim. In grassetto sono evidenziati gli angoli articolari che presentano differenza maggiore in termini di R^2.

Angoli articolari	**R^2**	**RMSE [°]**
hip rotation r	**0.7797**	0.0027
hip rotation l	**0.7194**	0.0023
hip flexion r	0.9955	0.0017
hip flexion l	0.9955	0.0017
hip adduction r	0.9941	0.0007
hip adduction l	0.9870	0.0009
knee angle r	0.9862	0.0031
knee angle l	0.9938	0.0023
ankle angle r	**0.6785**	0.0063
ankle angle l	**0.6968**	0.0059
lumbar extension	**0.7127**	0.0013
lumbar rotation	0.9981	5.3482
lumbar bending	0.9891	0.0007

Gli indici matematici (R^2 e RMSE) confermano l'analisi grafica dell'andamento degli angoli articolari. I problemi maggiori si riscontrano negli angoli di caviglia, nella rotazione dell'anca e nell'estensione lombare.

Dopo aver valutato la cinematica sono state analizzate le attivazioni muscolari generate dal tool *MocoTrack*, confrontandole con gli inviluppi lineari dei segnali EMG sperimentali (normalizzati al massimo valore osservato tra i dati a disposizione per il soggetto in analisi).

Le attivazioni generate da Moco per il movimento tracciato presentano un elevata variabilità rispetto ai dati di elettromiografia.
Il gluteo massimo (Figura 16) ha un andamento simile ai valori di EMG nella prima zona del ciclo del passo (fino al 15%), successivamente i valori di attivazione risultano maggiori e in corrispondenza del 60% del passo presenta un picco di attivazione. Il gluteo medio (Figura 16) presenta un andamento all'incirca costante sui valori di attivazione compresi tra lo 0.5 e 0.7, si discosta dai valori di EMG non presentando il picco iniziale di attivazione e valori maggiori nel restante ciclo del passo.
Il retto femorale (Figura 17) presenta un'attivazione iniziale maggiore rispetto ai valori di EMG e un picco non voluto in corrispondenza del 60% del ciclo del passo. Il vasto mediale (Figura 17) ha un profilo della curva simile al segnale elettromiografico ma con valori di attivazione inferiori.
Il gastrocnemio mediale (Figura 18) inizia con attivazioni maggiori rispetto ai dati di EMG e mantiene un picco di attivazione per una percentuale del ciclo del passo molto maggiore rispetto ai dati sperimentali. Il soleo (Figura 18) inizia e finisce con attivazioni maggiori rispetto all'EMG. Il tibiale anteriore (Figura 18) rimane sempre attivo, al contrario di quanto osservato sperimentalmente (in cui non risulta attivo durante quasi tutta la fase di appoggio).

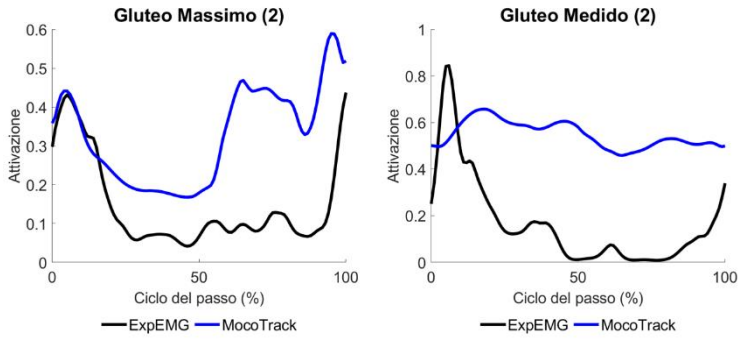

Figura 16: Confronto tra le attivazioni rilevate da EMG (nero) e quelle stimate da Moco (blu) per i muscoli: gluteo massimo (2) e gluteo medio (2).

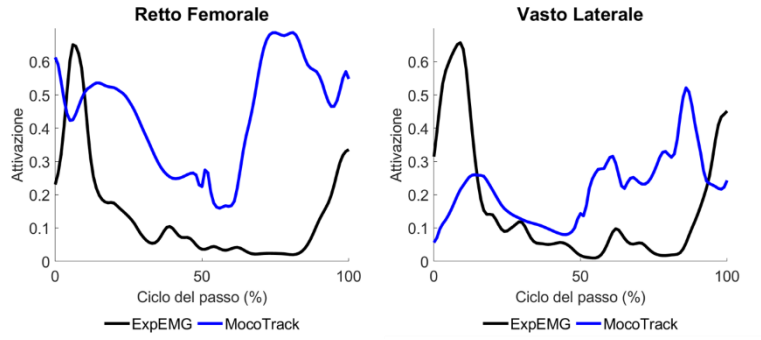

Figura 17: Confronto tra le attivazioni rilevate da EMG (nero) e quelle stimate da Moco (blu) per i muscoli: retto femorale e vasto laterale.

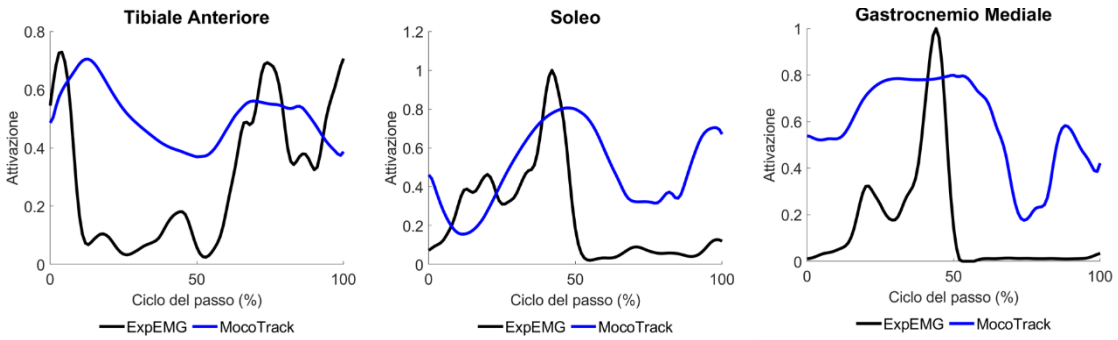

Figura 18: Confronto tra le attivazioni rilevate da EMG (nero) e quelle stimate da Moco (blu) per i muscoli: tibiale anteriore, soleo e gastrocnemio mediale.

Graficamente le attivazioni ottenute risultano molto differenti dai segnali elettromiografici, per verificare matematicamente l'analisi grafica eseguita nella tabella seguente vengono riportati i valori di R^2 e RMSE per confermare l'analisi grafica appena svolta.

Tabella 12: Valori di R^2 e RMSE per le attivazioni ottenute da MocoTrack a confronto con i dati di EMG. In grassetto sono evidenziati le attivazioni che presentano differenze maggiori in termini di R^2.

Muscoli	**R^2**	**RMSE**
Gluteo Medio (2)	**0.0541**	0.4246
Gluteo Massimo (2)	0.1626	0.2238
Retto Femorale	**0.0229**	0.3648
Vasto Laterale	**0.0204**	0.2293
Gastrocnemio Mediale	0.2881	0.4850
Soleo	**0.0848**	0.3670
Tibiale Anteriore	**0.0198**	0.3186

Per la risoluzione del problema di tracciamento con una convergenza ottimale da parte del solutore, i tempi computazioni relativi al computer (descritto in sezione 3.2.7 del capitolo 3) sono di 47 minuti e 47 secondi.

4.2 Step2: Analisi risultati della simulazione predittiva

L'analisi dei risultati prodotti dal tool *MocoPredict* ricalca la struttura mostrata per risultati del *MocoTrack*. Al contrario del *MocoTrack* sono state eseguite quattro simulazioni differenti di camminata i cui risultati verranno confrontati con il dato sperimentale, una ad una.

4.2.1 Analisi qualitativa della camminata predetta

L'analisi qualitativa delle camminate predette utilizzando diverse funzioni costo e obiettivo ha evidenziato come aggiungendo sotto task (*MocoGoals*) la cinematica del modello fosse migliorata progressivamente avvicinandosi sempre più ad una camminata 'normale'. In particolare, la flessione in avanti del busto e la breve fase di appoggio che caratterizzavano il risultato della prima simulazione (in cui era presente il solo *MocoAverageSpeedGoal*), si erano rispettivamente ridotta ed estesa con l'imposizione del vincolo di periodicità (simulazione 2). Aggiungendo infine due *MocoGoals* (*MocoControlGoal e MocoSumSquaredStateGoal),* nelle simulazioni 3 e 4, la cinematica del modello risultava maggiormente paragonabile ad una camminata normale.
Nella fase compresa tra lo 0% e il 50% col progredire delle simulazioni l'angolo di ginocchio risulta più esteso, per la gamba in appoggio, e più flesso nella gamba opposta. Tra il 50% e il 100% del passo non si osservano particolari differenze tra le simulazioni. L'inclinazione del busto all'istante finale (100% del ciclo del passo) è più marcata nella simulazione 1, mentre la fase di appoggio presenta un'ampiezza simile tra tutte le simulazioni.

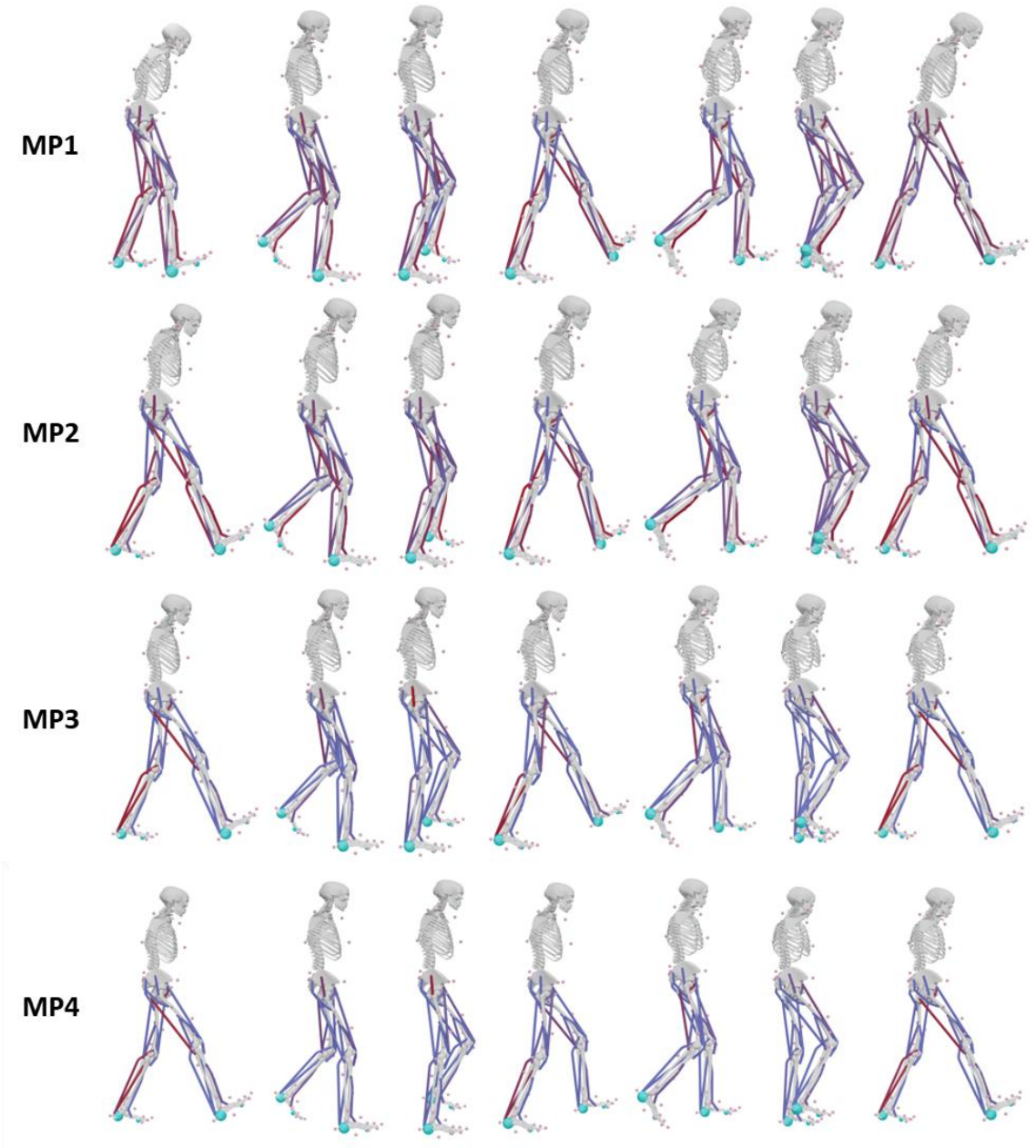

Figura 19: Frame relativi alle camminate predette da MocoPredict nelle varie simulazioni (MP1, MP2, MP3 e MP4). Le varie percentuali del ciclo del passo (es. 0%, 0-50%, 50%, 50-100% e 100%), predetto da Moco con il modello muscoloscheletrico scalato sul soggetto della KGC6, sono rappresentate attraverso frame.

4.2.2 Analisi grafica e statistica dei risultati predetti

Di seguito è riportato l'andamento degli angoli articolari predetti da Moco, confrontati con i valori ottenuti dalla simulazione di cinematica inversa in OpenSim (eseguita a partire dai dati della KGC6). Per ogni angolo articolare trattato sono riportati i grafici relativi alle quattro simulazioni (MP1, MP2, MP3 e MP4).

L'estensione lombare (Figura 20) predetta, per tutte e quattro le simulazioni, presenta valori maggiori rispetto alla cinematica stimata da OpenSim. L'angolo di flessione laterale (bending) (Figura 20) ha la maggior corrispondenza nella simulazione predittiva 2 (*MocoAverageSpeedGoal* e *MocoPeriodicityGoal*), mentre, per le simulazioni 3 e 4, tra 10% e 50% del ciclo del passo al posto di un picco massimo è presente un minimo, e viceversa tra il 50% e il 100%. L'angolo di rotazione per tutte le simulazioni presenta un andamento completamente differente rispetto ai dati di paragone.

Le simulazioni predittive presentano andamenti molto differenti rispetto ai risultati ottenuti dall'Inverse Kinematics per gli angoli di rotazione e adduzione dell'anca (Figura 21), mentre per l'angolo di flesso-estensione le curve presentano un profilo simile per tutte le simulazioni predittive; la maggior corrispondenza è relativa alla simulazione predittiva 2 in cui sono presenti i soli obiettivi di endpoint (*MocoAverageSpeedGoal* e *MocoPeriodicityGoal*).

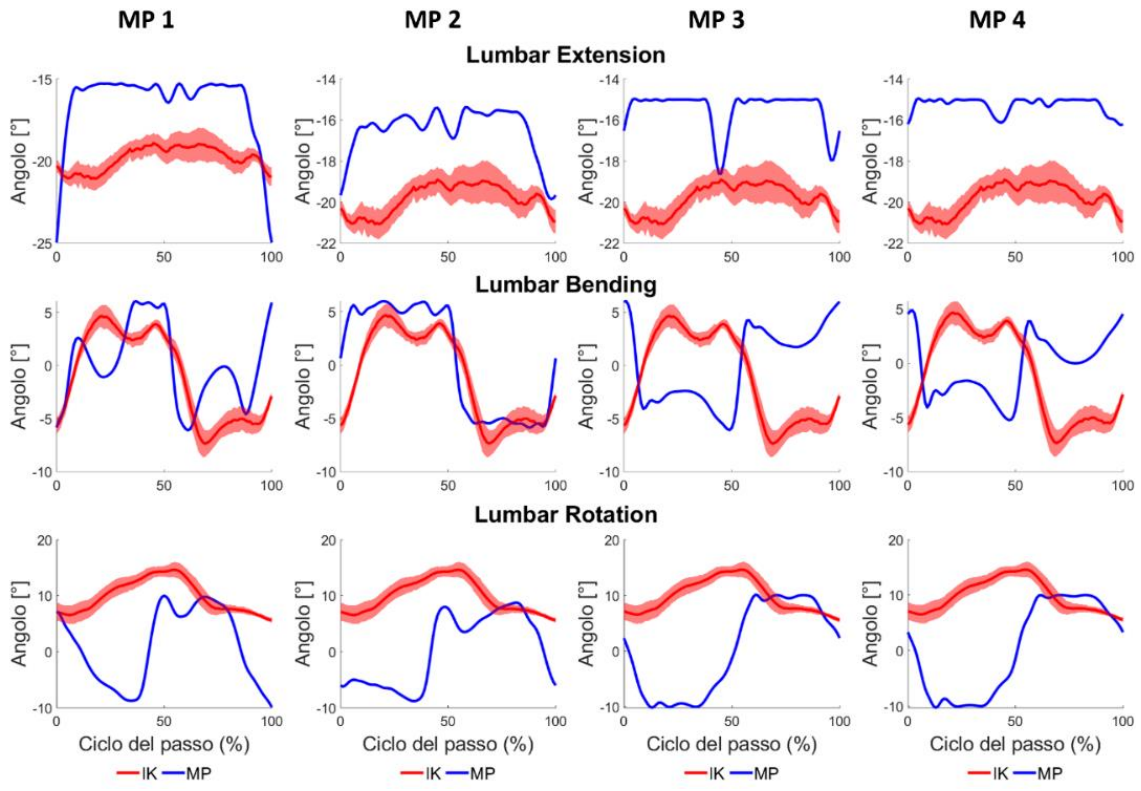

Figura 20: Confronto tra gli angoli articolari lombari di estensione, rotazione ed estensione laterale (bending) ottenuti dall'Inverse Kinematics (dati KGC6) di OpenSim (rosso) e predetti nelle vare simulazioni (MP) di Moco (blu).

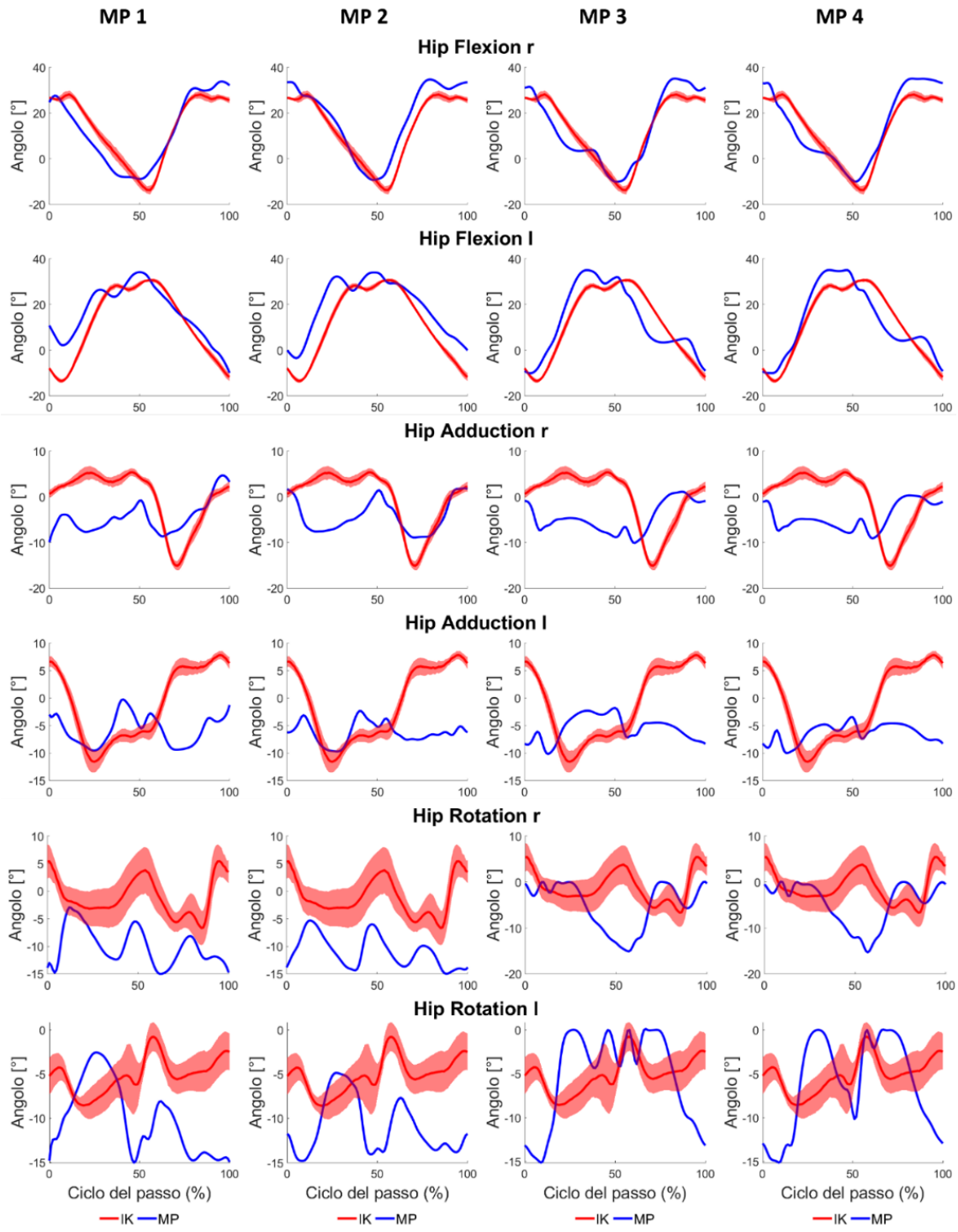

Figura 21: Confronto tra gli angoli articolari di flessione, adduzione e rotazione dell'anca ottenuti dall'Inverse Kinematics (dati KGC6) di OpenSim (rosso) e predetti nelle vare simulazioni (MP) di Moco (blu).

L'angolo di flesso-estensione di ginocchio (Figura 22) presenta un andamento coerente con i risultati di Inverse Kinematics, in particolar modo per la simulazione 1 che presenta il solo obiettivo di endpoint per la velocità media finale (*MocoAverageSpeedGoal*), mentre per la simulazione 3 e 4 nella fase di flessione (50% - 75%) sono presenti fasi intermedie di estensione. La fase finale del passo presenta un'ottima predizione della cinematica in quanto le curve si sovrappongono.

L'angolo di caviglia (Figura 22) presenta progressivamente nelle simulazioni un incremento non voluto di picchi di estensione e flessione rispetto ai risultati di confronto.

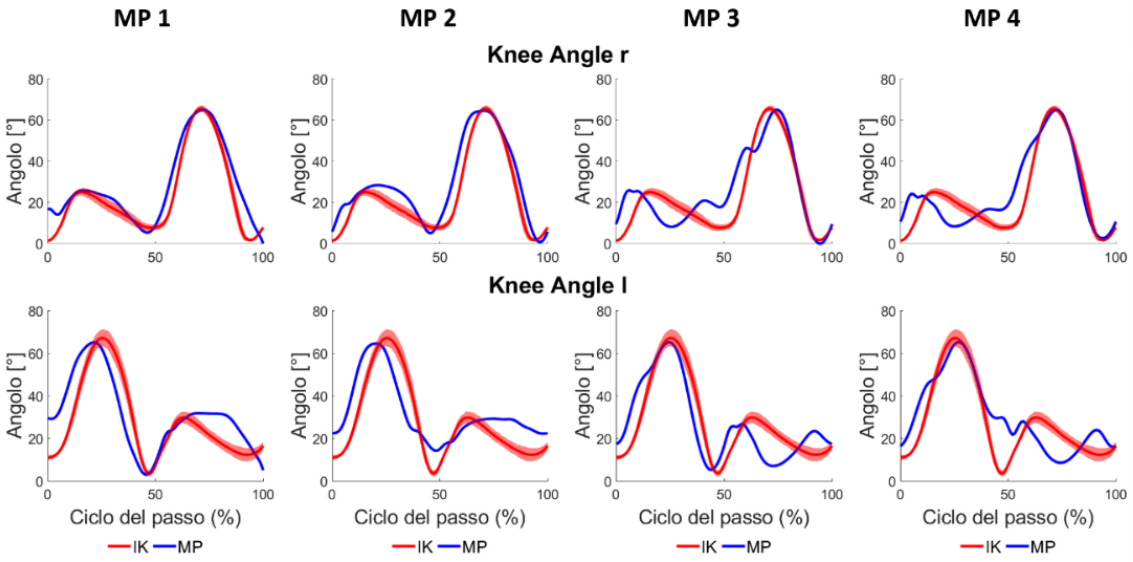

Figura 22: Confronto tra gli angoli articolari di flesso-estensione del ginocchio ottenuti dall'Inverse Kinematics (dati KGC6) di OpenSim (rosso) e predetti nelle vare simulazioni (MP) di Moco (blu).

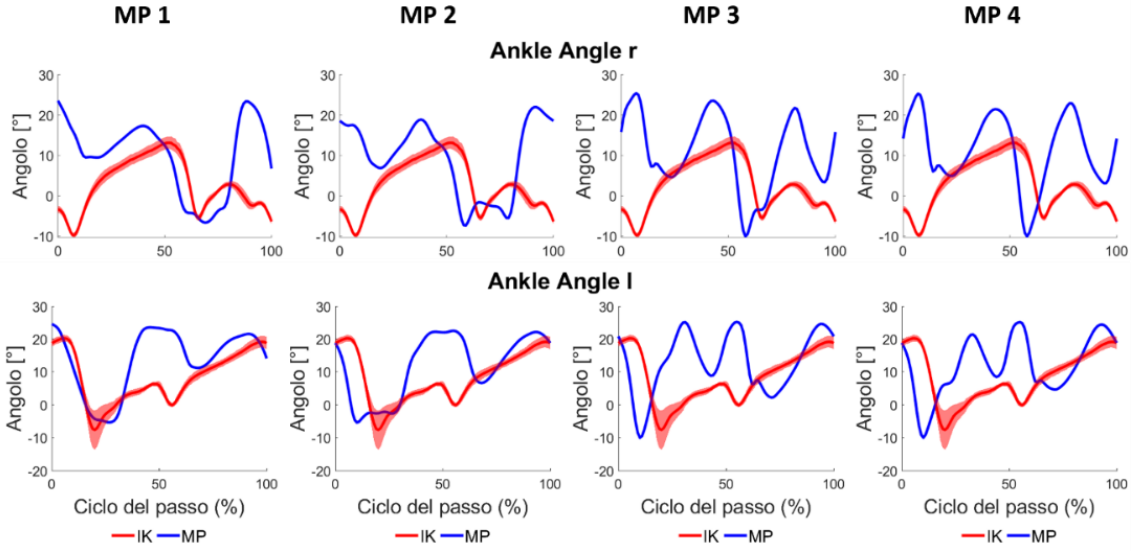

Figura 23: Confronto tra gli angoli articolari di flesso-estensione della caviglia ottenuti dall'Inverse Kinematics (dati KGC6) di OpenSim (rosso) e predetti nelle vare simulazioni (MP) di Moco (blu).

Nella Tabella 13 è riportato il valore di R^2 e RMSE per avere un confronto numerico.

Tabella 13:Valori di R^2 e RMSE per la cinematica ottenuta da MocoPredict nelle varie simulazioni (MP) a confronto con OpenSim. In grassetto sono evidenziati gli angoli articolari che presentano differenza minore in termini di R^2 nelle varie simulazioni (MP).

Angoli articolari	**R^2**				**RMSE [°]**			
	MP1	**MP2**	**MP3**	**MP4**	**MP1**	**MP2**	**MP3**	**MP4**
hip rotation r	0.00022	0.0128	**0.2836**	0.2397	0.0229	0.0240	0.0184	0.0176
hip rotation l	**0.3576**	0.2084	0.2514	0.0761	0.0128	0.0123	0.0118	0.0110
hip flexion r	0.8399	**0.9015**	0.8670	0.8409	0.0096	0.0082	0.0083	0.0091
hip flexion l	0.8466	**0.9088**	0.7784	0.8180	0.0105	0.0120	0.0105	0.0096
hip adduction r	0.0508	**0.2090**	0.0652	0.1213	0.0108	0.0097	0.0124	0.0121
hip adduction l	0.00047	0.0047	**0.2342**	0.0102	0.0137	0.0135	0.0142	0.0138
knee angle r	**0.9398**	0.9103	0.7906	0.8069	0.0104	0.0122	0.0140	0.0134
knee angle l	0.7364	**0.7438**	0.6904	0.5433	0.0139	0.0133	0.0146	0.0176
ankle angle r	0.0046	0.0111	0.00015	**0.0201**	0.0190	0.0195	0.0204	0.0210
ankle angle l	**0.2078**	0.0225	0.0496	0.0142	0.0138	0.0156	0.0180	0.0166
lumbar extension	**0.1257**	0.2411	0.0117	0.0248	0.0054	0.0046	0.0061	0.0063
lumbar rotation	**0.099**	0.0267	0.0210	0.0290	0.0171	0.0182	0.0204	0.0202
lumbar bending	0.2652	0.6184	**0.6720**	0.4855	0.005	0.0048	0.0095	0.0080

Dopo aver valutato la cinematica è necessario esaminare le attivazioni muscolari generate, nelle varie simulazioni, per il movimento predetto; nelle figure seguenti sono confrontate le attivazioni prodotte da *MocoPredict* con i dati sperimentali di EMG.

Le attivazioni predette presentano un miglioramento progressivo col complicarsi della funzione costo.

Le attivazioni predette per il gluteo massimo (Figura 24) risultano quasi nulle; al progredire delle simulazioni quella che rappresenta meglio l'andamento, anche se di valore molto inferiore, è la simulazione 1 (*MocoAveragesSpeedGoal*).

Il bicipite femorale lungo (Figura 25) si attiva e disattiva in maniera differente e non corrispondente rispetto al dato di EMG.

Il retto femorale (Figura 26) presenta un picco di attivazione inferiore rispetto al dato di EMG e traslato rispetto al ciclo del passo. Il vasto laterale e mediale (Figura 26) hanno valori di attivazione inferiori rispetto ai dati sperimentali, ma le zone di attivazione predette rispecchiano quelle dell'elettromiografia.

Il tibiale anteriore (Figura 27) presenta un profilo di attivazione simile al dato di elettromiografia. Il soleo e il gastrocnemio mediale hanno intervallo di attivazione nel ciclo del passo inferiore al progredire delle simulazioni.

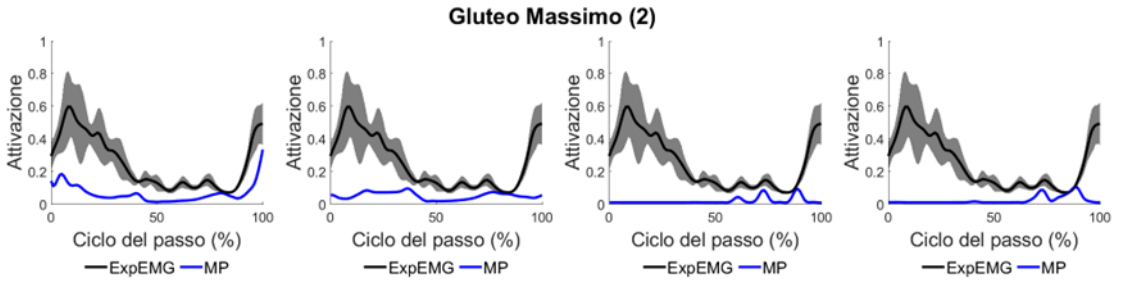

Figura 24: Confronto tra le attivazioni rilevate da EMG (nero) e quelle predette nelle varie simulazioni (MP) da Moco (nero) per il muscolo gluteo massimo (2).

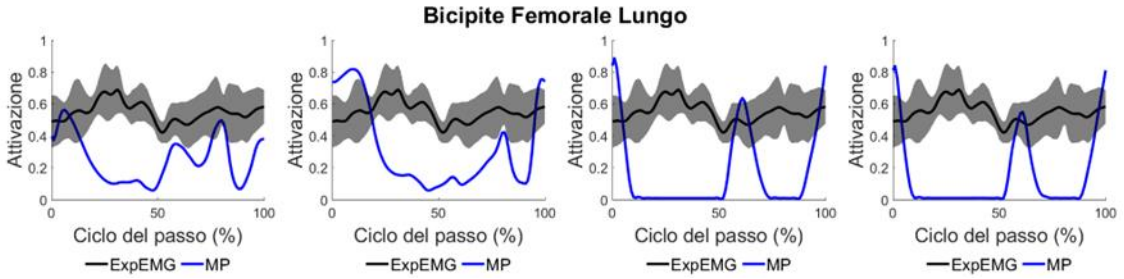

Figura 25: Confronto tra le attivazioni rilevate da EMG (nero) e quelle predette nelle varie simulazioni (MP) da Moco (nero) per il muscolo bicipite femorale lungo.

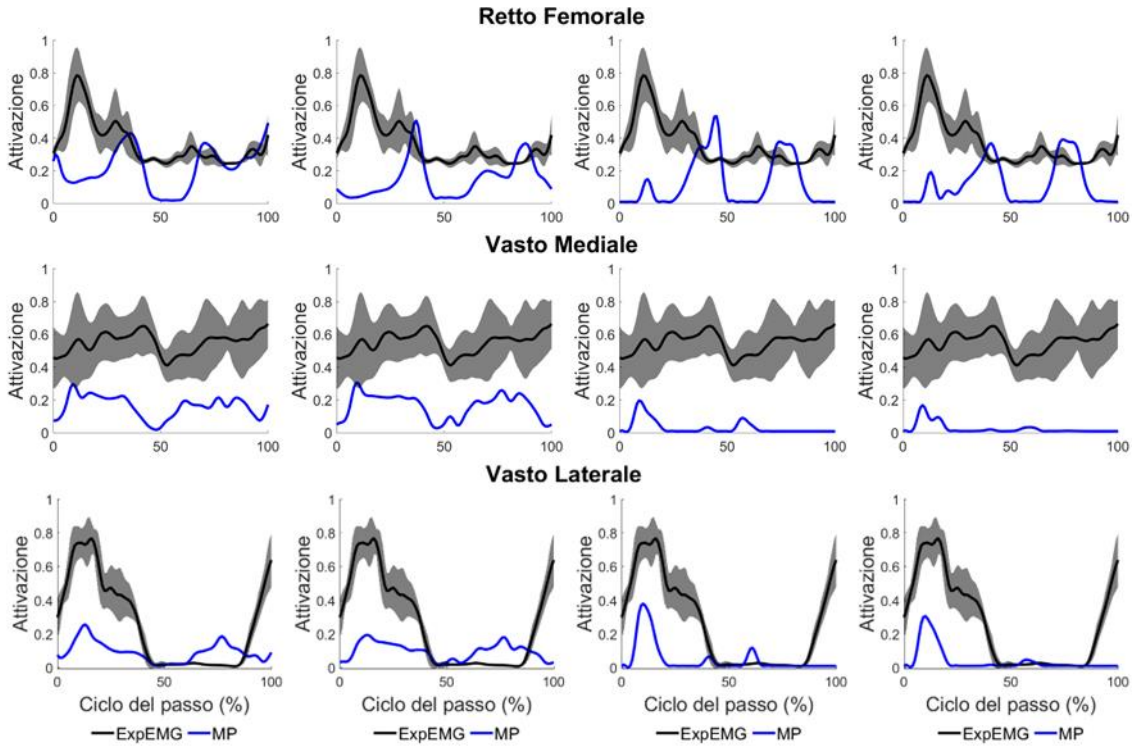

Figura 26: Confronto tra le attivazioni rilevate da EMG (nero) e quelle predette nelle varie simulazioni (MP) da Moco (nero) per i muscoli: retto femorale, vasto mediale e laterale.

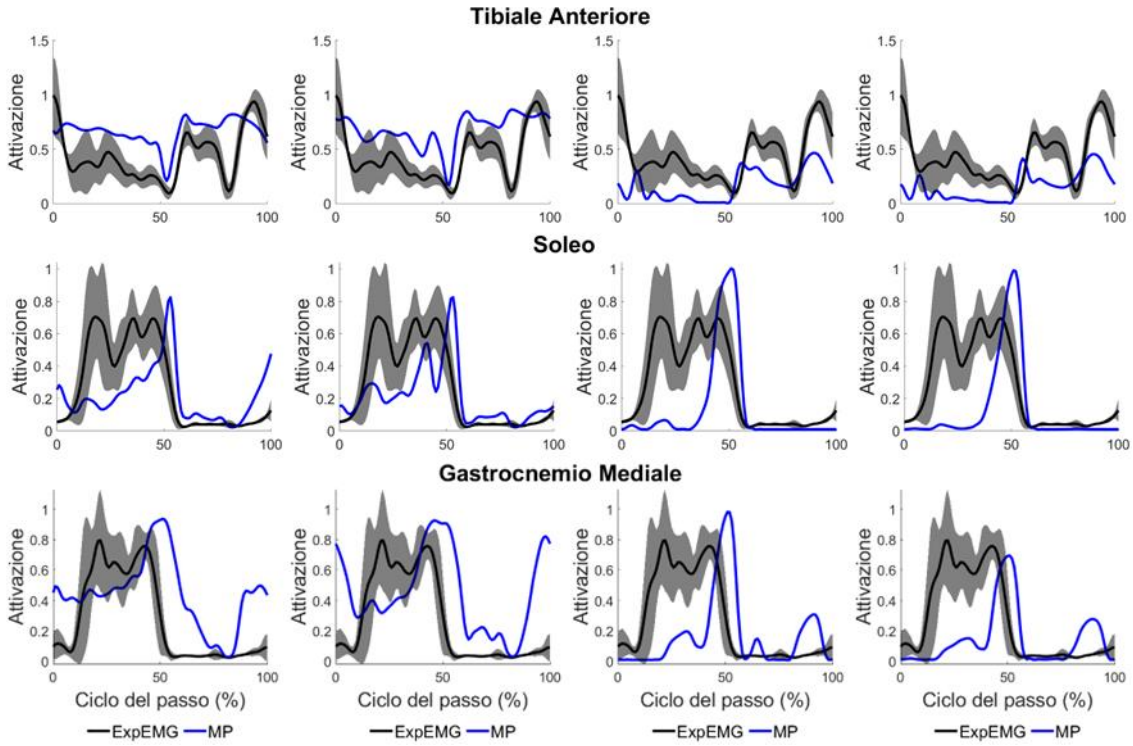

Figura 27: Confronto tra le attivazioni rilevate da EMG (nero) e quelle predette nelle varie simulazioni (MP) da Moco (nero) per i muscoli: tibiale anteriore, soleo e gastrocnemio mediale.

Nella Tabella seguente è riportato li valori di R^2 e RMSE per avere un confronto numerico.

Tabella 14: Valori di R^2 e RMSE per le attivazioni ottenute da MocoPredict nelle varie simulazioni predittive (MP) a confronto coni dati di EMG. In grassetto sono evidenziati le attivazioni che presentano differenza minore in termini di R^2 nelle varie simulazioni (MP).

Muscoli	**R^2**				**RMSE**			
	MP1	**MP2**	**MP3**	**MP4**	**MP1**	**MP2**	**MP3**	**MP4**
Gluteo max 2	0.2389	0.2652	**0.2964**	0.2952	0.0475	0.0490	0.0477	0.0478
Gluteo medio 2	0.4712	0.5103	0.5096	**0.5163**	0.0898	0.0861	0.0721	0.0773
Bicipite femorale lungo	0.3617	0.3734	0.4850	**0.4896**	0.0630	0.0447	0.0518	0.0571
Retto femorale	0.2534	0.3087	**0.3410**	0.3286	0.0314	0.0304	0.0318	0.0313
Vasto laterale	0.2985	0.3078	0.3175	**0.3239**	0.0289	0.0279	0.0309	0.0304
Vasto mediale	0.4140	0.4107	0.5357	**0.5398**	0.1216	0.1213	0.1250	0.1257
Tibiale anteriore	0.3302	0.3289	0.3583	**0.3740**	0.0512	0.04338	0.0461	0.0462
Soleo	0.2905	0.2628	0.35556	**0.3568**	0.0589	0.0583	0.0674	0.0652
Gastrocnemio mediale	0.3585	0.3974	**0.3921**	0.3626	0.0190	0.0214	0.0542	0.0601

Si nota che le simulazioni con incremento di complessità della funzione obiettivo hanno portato ad un miglioramento, anche se minimo, del valore di R^2 relativo alle attivazioni muscolari, mentre il valore del RMSE non presenta variazioni significative tra le varie simulazioni.

Per la risoluzione del problema di predizione con una convergenza ottimale da parte del solutore, i tempi computazioni relativi al computer (descritto in sezione 3.2.7 del capitolo 3) sono di 2 min e 48 sec per la simulazione 1, 1 min e 42 sec per la simulazione 2, 13 mi e 47 sec per la simulazione 3 e 31 mi e 33 sec per la quarta. Si può evincere che l'aggiunta di *Goals* da minimizzare ha portato ad un incremento dei tempi.

4.3 Analisi risultati delle ulteriori analisi svolte sui tools

4.3.1 Analisi grafica risultati di MocoInverse

L'analisi svolta del tool base, *MocoInverse*, ha evidenziato curve con andamento simile tra le varie simulazioni, con picco massimo e minimo di ampiezza differente. Con l'implementazione del tracciamento di dati elettromiografici, al variare del peso associato alla funzione (*MocoControlTrakingGoal*) le attivazioni muscolari stimate dal modello hanno mostrato un diverso livello di similarità con il dato sperimentale. In particolare, all'aumento del peso (1, 5, 10, 25, 50, 100), le curve generate da Moco risultavano con una corrispondenza maggiore rispetto ai profili sperimentali. Per maggiori dettagli, si rimanda ai grafici riportati in appendice.
I tempi computazionali per la risoluzione di *MocoInverse*, per entrambe le analisi, sono compresi tra 1 min e 5 min.

4.3.2 Analisi grafica MocoTrack

L'analisi svolta del tool base, *MocoTrack*, ha evidenziato che mesh temporali più fitte peggiorano la cinematica tracciata. L'aggiunta di un goal per il tracciamento dei dati elettromiografici ha portato ad un netto miglioramento dei pattern di attivazione muscolare predetti, ma ad un peggioramento della cinematica. In appendice vengono riportati i grafici relativi ai risultati delle due prove.

I tempi computazionali per la risoluzione di *Mocotrack* al variare dell'infittimento della mesh temporale sono aumentati, passando da un intervallo di 40-60 min, per l'infittimento minore (0.1) a 1-2h per l'infittimento maggiore (0.01).

Capitolo 5: Discussione

L’obiettivo del progetto di tesi era generare simulazioni predittive di camminata, con modelli muscoloscheletrici. Per fare questo si è usato Moco, software con cui è possibile realizzare simulazioni muscoloscheletriche mediante collocazione diretta. Moco è caratterizzato da un’interfaccia “user friendly” ed estremamente personalizzabile in base alle esigenze dell’utilizzatore; infatti, il linguaggio di programmazione rispecchia la struttura, già nota, di OpenSim. L’obiettivo del software è permettere la realizzazione di uno studio (*MocoStudy*), personalizzabile, per la risoluzione di problemi di optimal control. Inoltre, per semplificarne ulteriormente l’utilizzo, gli sviluppatori hanno realizzato tools preassemblati, ma comunque modificabili: *MocoInverse* e *MocoTrack*.
Essendo Moco molto recente, in quanto sviluppato nel 2017, ed essendo la tematica delle simulazioni predittive basata sulla risoluzione di problemi di optimal control con collocazione diretta ancora poco ricercata, la tesi ha avuto un carattere fortemente esplorativo, che ha visto dapprima lo studio e l'analisi dei due tools preassemblati (*MocoInverse* e *MocoTrack*) e infine l'utilizzo del *MocoPredict* per generare varie simulazioni predittive, modificando progressivamente la funzione obiettivo.

L’utilizzo dei due tools base è risultato semplice e intuitivo, essendo questi preassemblati e praticamente pronti per l’utilizzo.
Tramite *MocoInverse* sono state valutate le attivazioni generate e poi confrontate con i dati di EMG. Successivamente è nato l’interesse di implementare il tool introducendo un tracciamento dei dati di elettromiografia. Aumentando il peso dato a questa funzione (*MocoControlTrackingGoal*) i risultati di attivazione si avvicinavano sempre di più ai dati sperimentali.
Le analisi successive sono state svolte con *MocoTrack*. L’obiettivo era tracciare i soli dati sperimentali per generare la cinematica del soggetto in esame (KGC6). I risultati sono stati soddisfacenti, ma non per le attivazioni. Tuttavia, aggiungendo tra i termini di tracking anche i dati elettromiografici (*MocoControlTrackingGoal*), come prevedibile, questi sono migliorati (a scapito - in parte - della bontà di tracciamento della cinematica).

Giocando opportunamente con i pesi, così come tipicamente fatto utilizzando approcci EMG-assisted, verosimilmente si potrebbero ottenere risultati soddisfacenti.

Svolte le prime analisi per esplorare e personalizzare i tools, l'obiettivo principale è stato realizzare simulazioni predittive di camminata tramite uno studio personalizzato. I primi test (non riportati tra i risultati) non sono stati soddisfacenti, con i modelli che volavano o pattinavano (trascinando i piedi), evidenziando così la complessità del problema e la necessità di definire una condizione iniziale (più plausibile).
Per questo è stato realizzato un workflow caratterizzato da due step: (i) realizzazione di una condizione iniziale e (ii) realizzazione della simulazione predittiva.
La condizione iniziale, per semplicità nella struttura degli output, è stata realizzata tramite il tool *MocoTrack*; mentre la simulazione predittiva è stata realizzata con un *MocoStudy*, implementato progressivamente con *Goals* di complessità differente.

I risultati generati dal primo step del workflow, dal punto di vista cinematico, non presentano differenze evidenti tranne per l'angolo di flesso-estensione della caviglia a causa della gestione differente dei pesi imposti ai marker da Moco. Al contrario, le attivazioni muscolari generate si discostano dai dati di EMG, problema da considerare per le successive simulazioni predittive.

Nel secondo step si presupponeva un miglioramento generale progressivo nelle quattro simulazioni svolte. Così non è stato, in quanto le prime due simulazioni hanno portato a cinematiche con una migliore corrispondenza ai dati di confronto. Tuttavia, le attivazioni muscolari generate in questi primi due tentativi si discostavano marcatamente dai dati EMG sperimentali. L'aggiunta di termini/goals relativi all'attività muscolare (controlli e attivazioni) ha portato ad un netto miglioramento, associato ad una peggior predizione degli angoli articolari. Tutti i fattori sopra trattati sono fondamentali per la riuscita delle simulazioni così come l'utilizzo di ulteriori fattori e/o una diversa distribuzione dei pesi. Inoltre, va detto che alcune scelte potrebbero aver influenzato i risultati, tra cui il limitato numero di muscoli nel modello, la scelta di alcuni vincoli e/o controlli più associabili ad una camminata 'sana', così come l'uso di dati relativi a soggetti sani per definire la condizione iniziale. Studi futuri potrebbero vertere a ripetere il lavoro svolto su un dataset più ampio, in cui entrambi gli step sono informati da dati (di trials diversi) relativi allo

stesso soggetto. Questo potrebbe portare la simulazione predittiva a replicare la condizione iniziale, nel caso di soli vincoli di endpoint. Inoltre, l'uso di un maggior numero di contatti (sfere) poste in posizioni strategiche (per replicare un contatto reale) e l'utilizzo di modelli muscoloscheletrici più complessi (completi di arti superiori, articolazioni sbloccate e maggior numero di muscoli) potrebbe portare a un miglioramento in termini di cinematica e attivazioni, garantendo una maggiore stabilità nel movimento.

Grazie all'elevata flessibilità di personalizzazione e la possibilità di realizzare *Goals* specifici per ogni soggetto/paziente, l'uso di Moco può essere introdotto in ambito medico e fisioterapico per realizzare scenari ipotetici di diagnosi e terapie riabilitative attuabili sul paziente. Questo porterebbe a una riduzione dei tempi riabilitativi e permetterebbe di testare e valutare la terapia migliore per il problema dello specifico soggetto.

Un altro possibile utilizzo di Moco è in ambito sportivo. Essendo Moco basato sul Forward Approach, i movimenti repentini o impulsivi possono essere trattati meglio rispetto a quanto può fare un'analisi di ottimizzazione statica. Inoltre, con *Goals* personalizzati è possibile predire la traiettoria migliore per massimizzare le prestazioni in un determinato atto motorio.

Appendice

Script per la realizzazione della condizione iniziale

Figura 28: Script MocoTrack per generare la condizione iniziale.

Script per realizzazione delle simulazioni predittive

Figura 29: Scrip usato per le simulazioni predittive.

Risultati delle ulteriori analisi svolte sui tools di Moco

MocoInverse: confronto delle attivazioni tra Moco, Ottimizzazione Statica e EMG

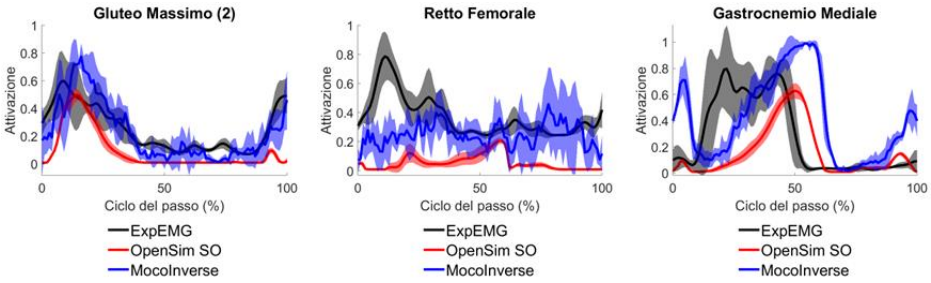

Figura 30: Confronto tra le attivazioni ottenute con MocoInverse (blu), Static Optimization (rosso) e i dati sperimentali di EMG (nero). Sono riportati i muscoli: gluteo massimo 2, retto femorale e gastrocnemio mediale. Il tool presenta le sole funzioni base (MocoControlGoal e MocoInitialActivationGoal).

MocoInverse: influenza del peso applicato al tracciamento dei dati di EMG

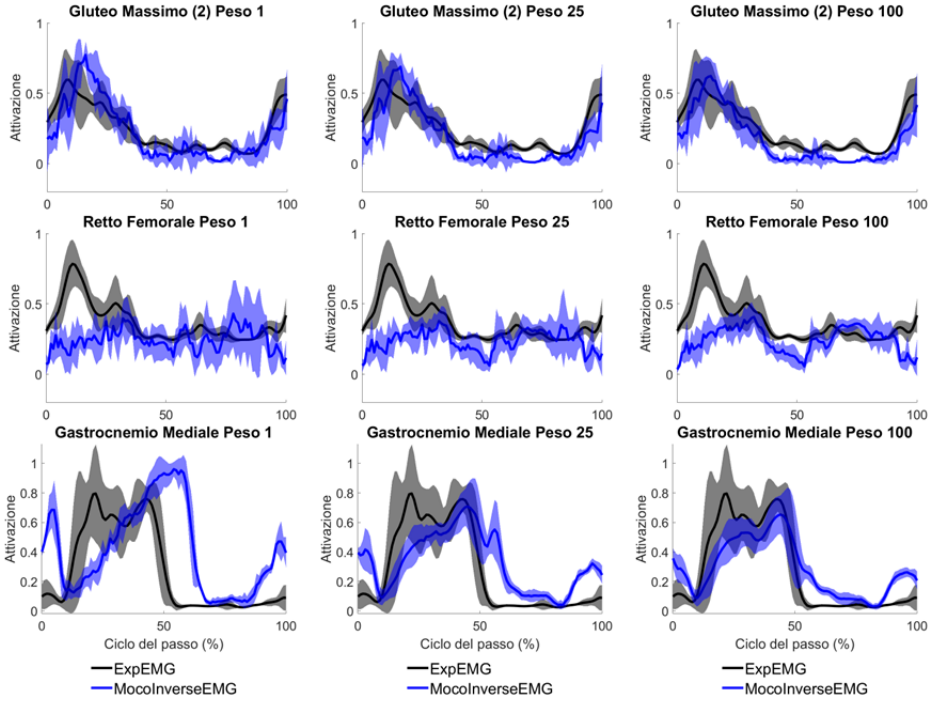

Figura 31: Confronto tra le attivazioni ottenute con MocoInverse (blu) e i dati sperimentali di EMG (nero). Sono riportati i muscoli: gluteo massimo 2, retto femorale e gastrocnemio mediale. Il tool MocoInverse oltre alle funzioni base (MocoControlGoal e MocoInitialActivationGoal) è implementato con MocoControlTrackingGoal per tracciare i dati di EMG a pesi differenti (1, 5, 10, 25, 50 e 100). Dai grafici si evince un avvicinamento delle attivazioni di Moco a quelle stimate da elettromiografico all'incremento del peso applicato alla funzione MocoControlTrackingGoal.

MocoTrack: influenza della mesh temporale sulla cinematica tracciata

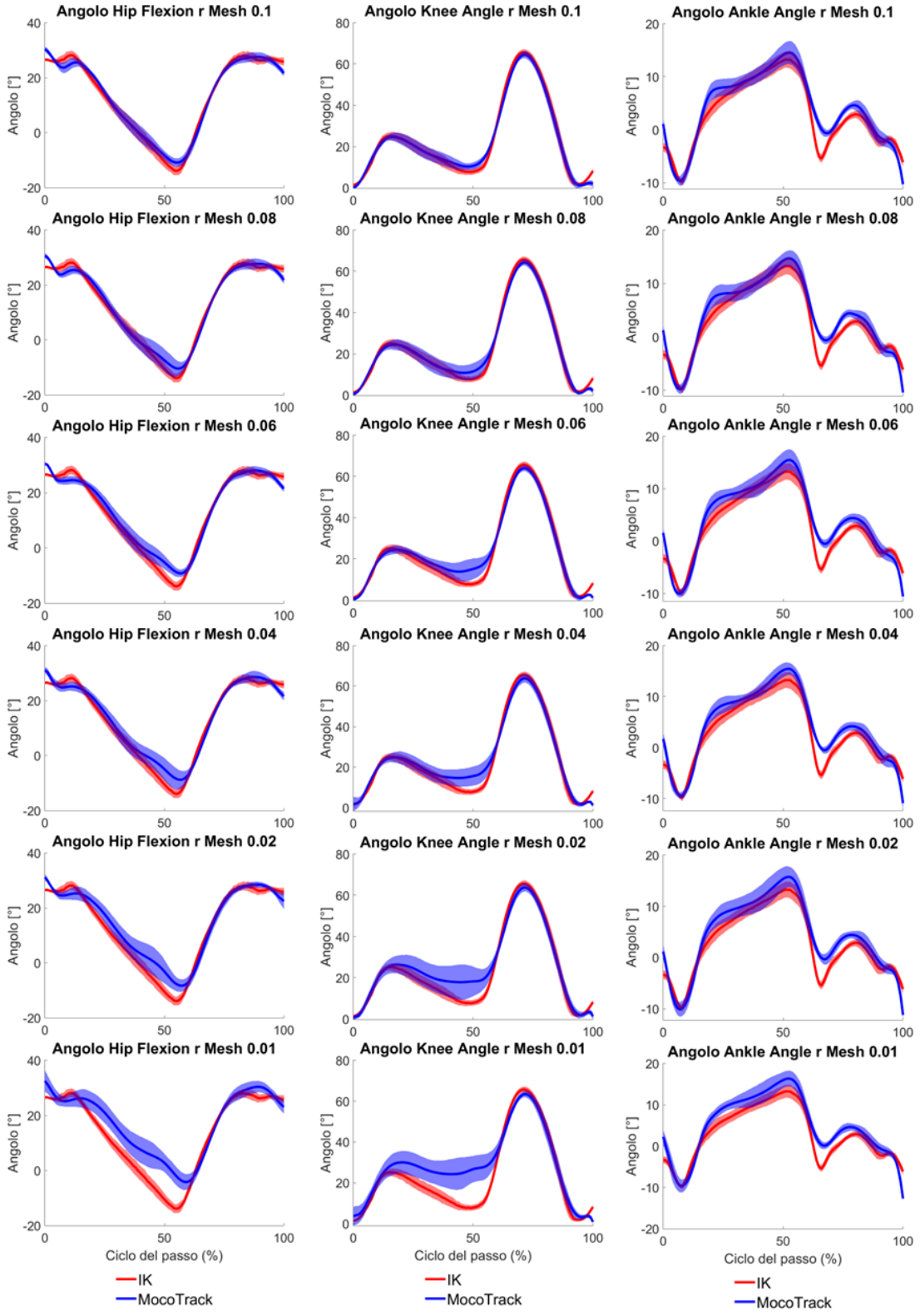

Figura 32: Confronto tra gli angoli articolari di flessione dell'anca, flesso-estensione del ginocchio e caviglia ottenuti con MocoTrack (blu) e OpenSim (rosso). Il tool presenta le sole funzioni base (MocomarkerTrackingGoal e MocoControlGoal) e vengono testate mesh temporali differenti.

MocoTrack: tracciamento cinematica e dati di EMG

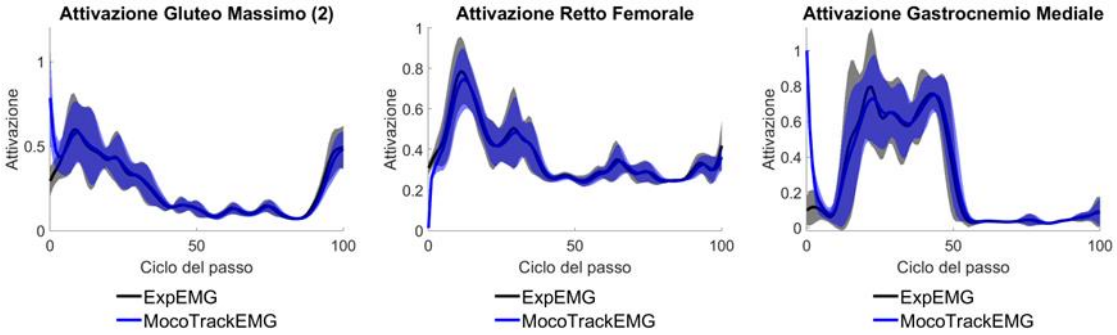

Figura 33: Confronto tra le attivazioni ottenute con MocoTrack (blu) e i dati sperimentali di EMG (nero). Il tool MocoTrack oltre alle funzioni base (MocomarkerTrackingGoal e MocoControlGoal) è implementato con MocoControlTrackingGoal per tracciare i dati di EMG. Il risultato è stato un tracciamento quasi identico ai rilevamenti di elettromiografia.

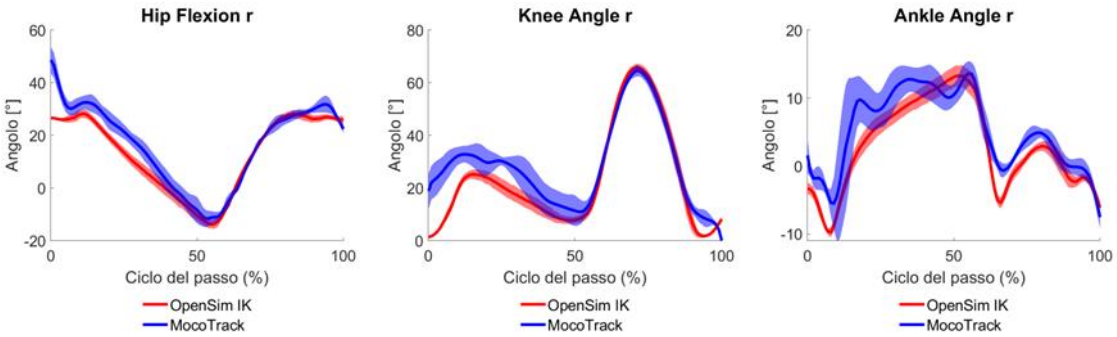

Figura 34: Confronto tra gli angoli articolari di flessione dell'anca, flesso-estensione del ginocchio e caviglia ottenuti con MocoTrack (blu) e OpenSim (rosso). Il tool MocoTrack oltre alle funzioni base (MocomarkerTrackingGoal e MocoControlGoal) è implementato con MocoControlTrackingGoal per tracciare i dati di EMG. Il risultato è stato un tracciamento non ottimale dei dati cinematici per via del peso elevato posto al tracciamento dei dati di elettromiografia.

Bibliografia

[1] R. Díaz, «Experimental Philosophy of Emotion», 2022.

[2] «PRIME PubMed | Ivan Petrovitch Pavlov, (1849-1936) conditioned reflexes». https://brain.unboundmedicine.com/medline/citation/4241320/Ivan_Petrovitch_Pavlov__1849_1936__conditioned_reflexes_ (consultato 3 marzo 2023).

[3] M. L. Latash, «5 - Control with forces and torques», in *Fundamentals of Motor Control*, M. L. Latash, A c. di San Diego: Academic Press, 2012, pp. 69–91. doi: 10.1016/B978-0-12-415956-3.00005-1.

[4] M. L. Latash, «6 - Control with muscle activations», in *Fundamentals of Motor Control*, M. L. Latash, A c. di San Diego: Academic Press, 2012, pp. 93–111. doi: 10.1016/B978-0-12-415956-3.00006-3.

[5] M. L. Latash, «7 - Control theory approaches», in *Fundamentals of Motor Control*, M. L. Latash, A c. di San Diego: Academic Press, 2012, pp. 113–124. doi: 10.1016/B978-0-12-415956-3.00007-5.

[6] A. Seireg e R. J. Arvikar, «A mathematical model for evaluation of forces in lower extremeties of the musculo-skeletal system», *J. Biomech.*, vol. 6, fasc. 3, pp. 313–326, mag. 1973, doi: 10.1016/0021-9290(73)90053-5.

[7] S. L. Delp, J. P. Loan, M. G. Hoy, F. E. Zajac, E. L. Topp, e J. M. Rosen, «An interactive graphics-based model of the lower extremity to study orthopaedic surgical procedures», *IEEE Trans. Biomed. Eng.*, vol. 37, fasc. 8, pp. 757–767, ago. 1990, doi: 10.1109/10.102791.

[8] T. K. Uchida, *Biomechanics of Movement: The Science of Sports, Robotics, and Rehabilitation*. MIT Press, 2021.

[9] D. T. Davy e M. L. Audu, «A dynamic optimization technique for predicting muscle forces in the swing phase of gait», *J. Biomech.*, vol. 20, fasc. 2, pp. 187–201, 1987, doi: 10.1016/0021-9290(87)90310-1.

[10] B. R. Umberger e R. H. Miller, «Optimal Control Modeling of Human Movement», in *Handbook of Human Motion*, B. Müller, S. I. Wolf, G.-P. Brueggemann, Z. Deng, A. McIntosh, F. Miller, e W. S. Selbie, A c. di Cham: Springer International Publishing, 2017, pp. 1–22. doi: 10.1007/978-3-319-30808-1_177-1.

[11] C. K. Chow e D. H. Jacobson, «Studies of human locomotion via optimal programming», *Math. Biosci.*, vol. 10, fasc. 3, pp. 239–306, apr. 1971, doi: 10.1016/0025-5564(71)90062-9.

[12] T. K. Ghosh e W. H. Boykin, «Analytic determination of an optimal human motion», *J. Optim. Theory Appl.*, vol. 19, fasc. 2, pp. 327–346, giu. 1976, doi: 10.1007/BF00934100.

[13] H. Hatze, «The complete optimization of a human motion», *Math. Biosci.*, vol. 28, fasc. 1, pp. 99–135, gen. 1976, doi: 10.1016/0025-5564(76)90098-5.

[14] «Publications dynamic programming». https://www.cs.utexas.edu/~shivaram/readings/b2hd-Bellman1957.html (consultato 3 marzo 2023).

[15] «L. S. Pontryagin, V. G. Boltyanskii, R. V. Gamkrelidze, E. F. Mishchenko, The mathematical theory of optimal processes | Control Processes Research Center». http://control.botik.ru/?publication_page=l-s-pontryagin-v-g-boltyanskii-r-v-gamkrelidze-e-f-mishchenko-the-mathematical-theory-of-optimal-processes (consultato 3 marzo 2023).

[16] M. G. Pandy, F. C. Anderson, e D. G. Hull, «A parameter optimization approach for the optimal control of large-scale musculoskeletal systems», *J. Biomech. Eng.*, vol. 114, fasc. 4, pp. 450–460, nov. 1992, doi: 10.1115/1.2894094.

[17] H. Celik e S. J. Piazza, «Simulation of aperiodic bipedal sprinting», *J. Biomech. Eng.*, vol. 135, fasc. 8, p. 81008, ago. 2013, doi: 10.1115/1.4024577.

[18] D. B. Leineweber, I. Bauer, H. G. Bock, e J. P. Schlöder, «An efficient multiple shooting based reduced SQP strategy for large-scale dynamic process optimization. Part 1: theoretical aspects», *Comput. Chem. Eng.*, vol. 27, fasc. 2, pp. 157–166, feb. 2003, doi: 10.1016/S0098-1354(02)00158-8.

[19] H. G. Bock e K. J. Plitt, «A Multiple Shooting Algorithm for Direct Solution of Optimal Control Problems*», *IFAC Proc. Vol.*, vol. 17, fasc. 2, pp. 1603–1608, lug. 1984, doi: 10.1016/S1474-6670(17)61205-9.

[20] M. Ackermann e A. J. van den Bogert, «Optimality principles for model-based prediction of human gait», *J. Biomech.*, vol. 43, fasc. 6, pp. 1055–1060, apr. 2010, doi: 10.1016/j.jbiomech.2009.12.012.

[21] *Solving Ordinary Differential Equations I*, vol. 8. Berlin, Heidelberg: Springer, 1993. doi: 10.1007/978-3-540-78862-1.

[22] E. Hairer e G. Wanner, *Solving Ordinary Differential Equations II*, vol. 14. Berlin, Heidelberg: Springer, 1996. doi: 10.1007/978-3-642-05221-7.

[23] M. Kelly, «An Introduction to Trajectory Optimization: How to Do Your Own Direct Collocation», *SIAM Rev.*, vol. 59, fasc. 4, pp. 849–904, gen. 2017, doi: 10.1137/16M1062569.

[24] C. L. Dembia, N. A. Bianco, A. Falisse, J. L. Hicks, e S. L. Delp, «OpenSim Moco: Musculoskeletal optimal control», *PLOS Comput. Biol.*, vol. 16, fasc. 12, p. e1008493, dic. 2020, doi: 10.1371/journal.pcbi.1008493.

[25] A. Seth *et al.*, «OpenSim: Simulating musculoskeletal dynamics and neuromuscular control to study human and animal movement», *PLOS Comput. Biol.*, vol. 14, fasc. 7, p. e1006223, lug. 2018, doi: 10.1371/journal.pcbi.1006223.

[26] J. T. Betts, *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*, Second. Society for Industrial and Applied Mathematics, 2010. doi: 10.1137/1.9780898718577.

[27] M. A. Sherman, A. Seth, e S. L. Delp, «Simbody: multibody dynamics for biomedical research», *Procedia IUTAM*, vol. 2, pp. 241–261, 2011, doi: 10.1016/j.piutam.2011.04.023.

[28] M. Posa, S. Kuindersma, e R. Tedrake, «Optimization and stabilization of trajectories for constrained dynamical systems», 2016.

[29] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, e M. Diehl, «CasADi: a software framework for nonlinear optimization and optimal control», *Math. Program. Comput.*, vol. 11, fasc. 1, pp. 1–36, mar. 2019, doi: 10.1007/s12532-018-0139-4.

[30] A. Wächter e L. T. Biegler, «On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming», *Math. Program.*, vol. 106, fasc. 1, pp. 25–57, mar. 2006, doi: 10.1007/s10107-004-0559-y.

[31] P. E. Gill, W. Murray, e M. A. Saunders, «SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization», *SIAM Rev.*, vol. 47, fasc. 1, pp. 99–131, gen. 2005, doi: 10.1137/S0036144504446096.

[32] C. Pizzolato *et al.*, «CEINMS: A toolbox to investigate the influence of different neural control solutions on the prediction of muscle excitation and joint moments during dynamic motor tasks», *J. Biomech.*, vol. 48, fasc. 14, pp. 3929–3936, nov. 2015, doi: 10.1016/j.jbiomech.2015.09.021.

[33] J. Cholewicki e S. M. McGill, «EMG assisted optimization: a hybrid approach for estimating muscle forces in an indeterminate biomechanical model», *J. Biomech.*, vol. 27, fasc. 10, Art. fasc. 10, ott. 1994, doi: 10.1016/0021-9290(94)90282-8.

[34] F. De Groote, A. L. Kinney, A. V. Rao, e B. J. Fregly, «Evaluation of Direct Collocation Optimal Control Problem Formulations for Solving the Muscle Redundancy Problem», *Ann. Biomed. Eng.*, vol. 44, fasc. 10, pp. 2922–2936, ott. 2016, doi: 10.1007/s10439-016-1591-9.

[35] B. I. Prilutsky e V. M. Zatsiorsky, «Optimization-based models of muscle coordination», *Exerc. Sport Sci. Rev.*, vol. 30, fasc. 1, pp. 32–38, gen. 2002, doi: 10.1097/00003677-200201000-00007.

[36] C. Schreiber e F. Moissenet, «A multimodal dataset of human gait at different walking speeds established on injury-free adult participants», *Sci. Data*, vol. 6, fasc. 1, Art. fasc. 1, lug. 2019, doi: 10.1038/s41597-019-0124-4.

[37] B. J. Fregly *et al.*, «Grand challenge competition to predict in vivo knee loads», *J. Orthop. Res. Off. Publ. Orthop. Res. Soc.*, vol. 30, fasc. 4, pp. 503–513, apr. 2012, doi: 10.1002/jor.22023.

[38] A. Esrafilian, L. Stenroth, M. E. Mononen, P. Tanska, J. Avela, e R. K. Korhonen, «EMG-Assisted Muscle Force Driven Finite Element Model of the Knee Joint with Fibril-Reinforced Poroelastic Cartilages and Menisci», *Sci. Rep.*, vol. 10, p. 3026, feb. 2020, doi: 10.1038/s41598-020-59602-2.

[39] A. Rajagopal, C. L. Dembia, M. S. DeMers, D. D. Delp, J. L. Hicks, e S. L. Delp, «Full-Body Musculoskeletal Model for Muscle-Driven Simulation of Human Gait», *IEEE Trans. Biomed. Eng.*, vol. 63, fasc. 10, pp. 2068–2079, ott. 2016, doi: 10.1109/TBME.2016.2586891.

[40] E. M. Arnold, S. R. Ward, R. L. Lieber, e S. L. Delp, «A model of the lower limb for analysis of human movement», *Ann. Biomed. Eng.*, vol. 38, fasc. 2, pp. 269–279, feb. 2010, doi: 10.1007/s10439-009-9852-5.

[41] S. R. Hamner, A. Seth, e S. L. Delp, «Muscle contributions to propulsion and support during running», *J. Biomech.*, vol. 43, fasc. 14, pp. 2709–2716, ott. 2010, doi: 10.1016/j.jbiomech.2010.06.025.

[42] «MOtoNMS: A MATLAB toolbox to process motion data for neuromusculoskeletal modeling and simulation | Source Code for Biology and Medicine | Full Text». https://scfbm.biomedcentral.com/articles/10.1186/s13029-015-0044-4 (consultato 5 marzo 2023).

[43] M. Millard, T. Uchida, A. Seth, e S. L. Delp, «Flexing Computational Muscle: Modeling and Simulation of Musculotendon Dynamics», *J. Biomech. Eng.*, vol. 135, fasc. 2, feb. 2013, doi: 10.1115/1.4023390.

[44] G. Serrancolí *et al.*, «Subject-Exoskeleton Contact Model Calibration Leads to Accurate Interaction Force Predictions», *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 27, fasc. 8, pp. 1597–1605, ago. 2019, doi: 10.1109/TNSRE.2019.2924536.