POLITECNICO DI MILANO
Scuola di Ingegneria Industriale e dell'Informazione
Corso di Laurea Specialistica in Ingegneria Informatica

Impact of muscle redundancy and nonlinearities on the muscle synergy hypothesis: a computational investigation

Relatore: Prof.ssa Giuseppina Gini
Correlatori: Dott. Cristiano Alessandro, PhD
Dott. Juan Pablo Carbajal, PhD
Prof. Robert Riener

Tesi di Laurea di:
Robin Urselli
Matricola 709970

Anno Accademico 2014-2015

For Jacopo and Iris...

Abstract

Musculoskeletal systems feature many degrees of freedom and muscles, which are characterized by complex nonlinear dynamics. One of the long-standing question in neuroscience is how the central nervous system (CNS) is able to efficiently and harmoniously perform motor control, despite this very high complexity. Researchers have tried to give an answer to this question formulating the muscle synergy hypothesis. Muscle synergies are groups of muscle activation patterns which can be linearly combined in order to generate complex muscle actuations. This strategy may simplify motor control because the CNS only has to choose how to combine these predefined patterns instead of synchronizing the individual muscles independently.

The classical approach in order to verify the muscle synergy hypothesis has been to record EMG (Electromyographic) signals during the execution of tasks, and try to extract components (i.e. synergies) able to reconstruct the EMG dataset. This approach provided evidence for the existence of synergies in many biological systems (frogs, cats, humans etc.). In particular, in humans, evidence of muscle synergies were found during postural balance, cycling, walking exercises etc. This approach leaves, however, many questions open. For example we don't know if it is theoretically possible to control a musculoskeletal system by linear combination of synergies, nor we know how the biomechanical properties of the system influence the synergy hypothesis. A possible computational approach to face these questions is to define a dynamical system and a set of desired tasks, and to see if it is possible to find a set of synergies, which, when linearly combined, are able to solve the desired tasks.

In this thesis we want to understand if the control by linear combinations of synergies is possible (question partially addressed in previous research), and to investigate the impact of redundancy and muscle nonlinearities (two important biomechanical features of musculoskeletal systems) on the synergy hypothesis. We modeled the human arm as a 2-joint kinematic chain, we synthesized appropriate synergies, and we measured their performance in

solving reaching tasks. We incrementally added the biomechanical features mentioned above: first we actuated the kinematic chain by means of torques applied to the joints. This model is not redundant as the number of inputs is equal to the number of joints; then we actuated the kinematic chain by means of 6 forces applied to the links, thus introducing redundancy as the number of forces exceeds the number of joints; finally we generated the 6 forces applied to the links by means of nonlinear muscles controlled in activation, thus introducing muscle nonlinearities. We were able to control the torque actuated and the force actuated kinematic chain obtaining the same performance with an equal number of synergies. We have found, instead, that the activation actuated kinematic chain requires a higher number of synergies in order to obtain similar performance.

Our results suggest that redundancy does not necessarily increase the number of synergies required to execute a task, and therefore that the number of muscles may not affect the complexity of the control. Nonlinearities, coming for example from muscle dynamics, may instead make the job of the CNS harder, and require a higher number of synergies.

Sommario

Il nostro apparato locomotore è estremamente complesso ed è composto da molti giunti, moltissime ossa e ancor più muscoli. Nonostante tutta questa complessità riusciamo a svolgere una quantità di azioni in modo spontaneo e armonioso, senza che ci sembri di compiere il minimo sforzo. Una delle domande nel campo neuroscientifico che più occupa i ricercatori è come faccia quindi il sistema nervoso a controllare il nostro apparato locomotore. Tra i molti fattori che ne rendono il controllo estremamente complesso ve ne sono due che hanno un ruolo fondamentale: la ridondanza e le non linearità. La ridondanza è caratterizzata dal fatto che il numero di muscoli supera di gran lunga quello dei gradi di libertà, per cui il sistema nervoso si trova a dover scegliere tra un numero infinito di modi di eseguire un task. La non linearità, invece, implica che combinazioni lineari di input non risultano in combinazioni lineari di output. Task molto simili potrebbero richiedere attuazioni molto diverse per essere eseguiti e, come conseguenza, il sistema nervoso non può utilizzare la sua esperienza passata per generare comandi appropriati.

Per eseguire un task, il sistema nervoso deve generare i segnali da mandare ai muscoli (i.e. attivazione muscolare), in modo tale che questi generino le forze che producono il movimento desiderato. In un approccio classico, il sistema nervoso dovrebbe generare i segnali individualmente per ogni muscolo, sincronizzandoli in maniera tale da ottenere l'effetto desiderato. Una delle ipotesi formulate dai ricercatori per cercare di spiegare come il sistema nervoso riesca a dominare questa complessità è quella delle sinergie muscolari.

Sinergie muscolari

L'ipotesi delle sinergie muscolari è una specializzazione dell'ipotesi del controllo motorio basato su architetture modulari già formulata da Nikolai Bernstein molti anni addietro. Le sinergie muscolari sono schemi di coattivazione muscolare, ogni sinergia descrive cioè uno specifico schema di

co-attivazione tra più muscoli. Al fine di eseguire un task, il sistema nervoso combina linearmente queste sinergie in modo da ottenere le complesse attivazioni muscolari necessarie per eseguire il task. Vi è quindi, in termini del numero di variabili da controllare, una notevole riduzione della dimensionalità del problema. Ciò risulta in una semplificazione del controllo, perché il sistema nervoso deve solo decidere come combinare le sinergie.

L'aspetto più importante nella teoria delle sinergie muscolari è che sono invarianti rispetto al task. Con ciò vogliamo dire che il sistema nervoso usa, almeno in parte, le stesse sinergie per eseguire task diversi [24, 18, 17, 25]. Alcuni ricercatori hanno anche scoperto che le sinergie potrebbero essere invarianti rispetto ai soggetti, nel senso che sinergie molto simili sono state trovate in individui diversi della stessa specie [23, 30, 25].

Sono stati proposti diversi modelli per le sinergie muscolari (si veda [2] per un sommario completo sui diversi modelli), tra questi quello delle sinergie tempo-varianti che useremo anche in questa tesi. Formalmente, le sinergie muscolari sono funzioni tempo-varianti $\mathbf{w}_j$ di attivazioni muscolari e vengono modulate in ampiezza e shiftate nel tempo da coefficienti a_j e τ_j. L'attivazione muscolare totale $\boldsymbol{u}(t)$ è data da:

$$\boldsymbol{u}(t) = \sum_{j=1}^{n} a_j \mathbf{w}_j(t - \tau_j) \tag{1}$$

dove n è il numero di sinergie.

L'approccio classico per verificare o confutare l'ipotesi delle sinergie muscolari è quello di registrare segnali EMG (elettromiografici) durante l'esecuzione di task e usare tecniche di analisi dei dati (PCA, ICA, NMF etc.) per estrarre componenti in grado di ricostruire il dataset di segnali EMG. Sugli umani questo approccio è stato applicato con successo in diversi studi su task di reaching [23, 21, 8, 20, 7, 19, 13], pedalata [42, 25], camminata [43, 17, 25], corsa [14], mantenimento dell'equilibrio [63, 18, 17]. L'approccio basato su analisi di EMG presenta tuttavia il problema che i risultati non possono essere facilmente validati (non si possono usare le sinergie estratte per verificare se il task viene eseguito correttamente), perciò negli ultimi anni sono emersi modelli che tentano di integrare nell'analisi le proprietà del task [18, 62, 27, 28]. Nonostante questi sforzi, rimane il problema che il vero rapporto tra input (sinergie muscolari) e output (cinematica), rappresentato dalla dinamica dell'apparato locomotore viene considerato molto raramente [53, 10, 2]. In particolare non è chiaro se è possibile controllare l'apparato locomotore tramite combinazione lineare di sinergie e non è noto se fattori biomeccanici come la ridondanza e le non linearità influenzano l'ipotesi delle sinergie muscolari.

Obiettivi

L'obiettivo di questa tesi (sezione 1.4) è capire (1) come ridondanza e non linearità influenzano l'ipotesi delle sinergie muscolari e (2) se è possibile controllare un sistema muscolo-scheletrico che presenta queste proprietà tramite combinazione lineare di sinergie.

Metodi

Per rispondere alle domande di cui sopra, utilizziamo un approccio computazionale, definendo un modello di braccio umano a due gradi di libertà e aggiungendo sistematicamente ad uno ad uno i fattori biomeccanici di interesse (i.e. ridondanza e non linearità). In una prima fase attuiamo il braccio con momenti applicati ai giunti (sezione 2.2.3); successivamente utilizziamo 6 forze applicate ai link, introducendo quindi la ridondanza (sezione 2.2.4); infine generiamo le forze tramite muscoli non lineari controllati in attivazione, introducendo quindi la non linearità (sezione 2.2.4). Per ognuno di questi modelli sintetizziamo delle sinergie e valutiamo le prestazioni durante l'esecuzione di task di reaching. Le sinergie sono definite nello spazio dell'input del modello via via analizzato (momenti, forze, attivazioni). Le prestazioni sono valutate definendo per ogni modello più di 800 task di reaching uniformemente distribuiti nello spazio operativo del braccio e calcolando l'errore ottenuto approssimando l'attuazione desiderata che risolve il task con una combinazione lineare delle sinergie. Per avere una misura di come l'errore scala con l'aumentare delle sinergie, abbiamo valutato le prestazioni per un numero di sinergie da 1 a 12.

Per rappresentare e sintetizzare sinergie utilizziamo DRD (Dynamic Response Decomposition) [2] (sezione 2.1), un metodo generale per trovare soluzioni a task come combinazione lineare di primitive, partendo da una traiettoria desiderata nello spazio del task. Il metodo sfrutta la dinamica inversa del sistema per calcolare la soluzione desiderata a partire dalla traiettoria e successivamente approssima la soluzione con le sinergie. Lo stesso metodo viene anche utilizzato per sintetizzare iterativamente sinergie (*reduction*) a partire dalle risposte del sistema ad un set rappresentativo di input (*exploration*) (sezione 2.3).

Il modello di braccio umano utilizzato è costituito da una catena cinematica planare, con due giunti rotazionali rappresentanti il gomito e la spalla. I dati antropometrici sono stati presi dalla letteratura [52].

Modello attuato da momenti ai giunti (sezione 2.2.3) Il modello della catena cinematica attuata da momenti applicati ai giunti è descritto dalle

ben note equazioni del moto:

$$\boldsymbol{D}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{E}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q}) = \boldsymbol{\tau} \tag{2}$$

dove $\boldsymbol{q}$ rappresenta lo stato del sistema (i.e. posizione dei giunti)e $\boldsymbol{\tau}$ i momenti applicati ai giunti. Le matrici $\boldsymbol{D}$, $\boldsymbol{E}$ e $\boldsymbol{G}$ rappresentano i termini inerziali, centrifughi e di coriolis, e gravitazionali. In questo modello le sinergie sono funzioni 2-dimensionali del tempo e rappresentano momenti ai giunti.

Modello attuato da forze (sezione 2.2.4) In questo modello la catena cinematica è attuata da 6 forze applicate ai link, rappresentanti 4 muscoli monoarticolari e 2 muscoli biarticolari. Il modello è ridondante perché ci sono più forze che gradi di libertà. Le forze $\boldsymbol{F}$ generano il momento ai giunti tramite i bracci fissi della matrice $\boldsymbol{L}$ (Leverarm). I parametri muscolari (punti di origine/inserimento, braccio della forza, forza massima etc.) sono stati stimati tramite una procedura di lumping [52] a partire dai dati di 19 muscoli del braccio umano (appendice A). Il sistema è descritto dall'equazione:

$$\boldsymbol{D}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{E}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q}) = \boldsymbol{\tau} = \boldsymbol{LF} \tag{3}$$

In questo modello le sinergie sono funzioni 6-dimensionali del tempo e rappresentano forze esercitate sui link dai muscoli.

Modello attuato da muscoli (sezione 2.2.5) In questo modello le 6 forze del modello precedente sono generate da muscoli non lineari. Ogni muscolo è composto da un elemento contrattile (CE) ed un elemento passivo (PE). L'elemento contrattile genera la forza dipendentemente dall'attivazione muscolare, dalla velocità di contrazione (force-velocity relation) e dalla lunghezza del muscolo (force-length relation). L'elemento passivo modella le proprietà elastiche passive del muscolo e genera una forza dipendente dalla sua lunghezza (force-length relation). La lunghezza dei muscoli è calcolata a partire dalla posizione della catena cinematica. La relazione tra forza generata, velocità di contrazione e lunghezza del muscolo è fortemente non lineare (Figure 2.7 e 2.8). Le equazioni che descrivono il moto del sistema sono:

$$\boldsymbol{D}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{E}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q}) = \boldsymbol{LC}(\boldsymbol{l}, \dot{\boldsymbol{l}})\boldsymbol{M} + \boldsymbol{LP}(\boldsymbol{l}) \tag{4}$$

dove $\boldsymbol{M}$ è l'attivazione muscolare, $\boldsymbol{C}(\boldsymbol{l}, \dot{\boldsymbol{l}})$ e $\boldsymbol{P}(\boldsymbol{l})$ incapsulano le caratteristiche non lineari dei muscoli ed $\boldsymbol{l}$ è il vettore che rappresenta lo stato (i.e. lunghezza) dei muscoli. Quindi, la relazione tra forza generata e attivazione

non solo è non lineare, ma dipende anche dallo stato (i.e. posizione e velocità) della catena cinematica. In questo modello le sinergie sono funzioni 6-dimensionali del tempo e rappresentano attivazioni muscolari.

Risultati

Modello attuato da momenti ai giunti (capitolo 3) In questo modello, con 8 sinergie definite nello spazio dei momenti, si è potuto effettuare task di reaching in tutto lo spazio operativo con un errore medio in posizione di $1mm$. Questi risultati sono simili a quelli già ottenuti precedentemente in [2].

Modello attuato da forze (capitolo 4) Nella catena cinematica attuata da forze non solo ci troviamo a dover affrontare il problema della ridondanza, ma anche quello di vincoli biologici dovuti al fatto che i muscoli possono generare solo forze positive. Al fine di calcolare le forze desiderate necessarie per eseguire una traiettoria di reaching, dobbiamo trovare un modo per calcolare le forze a partire dai momenti ai giunti. Per via della ridondanza, tuttavia, esistono infinite forze $\boldsymbol{F}$ che producono gli stessi momenti ai giunti $\boldsymbol{\tau}$. La relazione generale per calcolare la forza $\boldsymbol{F}$ dai momenti è data da:

$$\boldsymbol{F} = \boldsymbol{L}^{+}\boldsymbol{\tau} + \boldsymbol{N}\boldsymbol{w} \tag{5}$$

dove $\boldsymbol{L}^{+}$ è la matrice pseudo-inversa di $\boldsymbol{L}$ ed $\boldsymbol{N}$ è una base ortogonale di $\ker(\boldsymbol{L})$. Il termine $\boldsymbol{w}$ è il cosiddetto *nullshift* e può essere usato per variare il profilo della forza $\boldsymbol{F}$ senza modificare il momento risultante, poiché $\boldsymbol{N}\boldsymbol{w} \in \ker(\boldsymbol{L})$. Abbiamo utilizzato il nullshift $\boldsymbol{w}$ per calcolare delle forze desiderate che fossero positive e abbiamo dimostrato che se tutte le forze sono calcolate con un unico nullshift $\boldsymbol{w}^{*}$ uguale per tutte, la dimensionalità dell'insieme delle forze $\boldsymbol{F}$ rispetto a quella dell'insieme dei momenti ai giunti $\boldsymbol{\tau}$ viene conservata. Abbiamo sintetizzato un set di sinergie e abbiamo dimostrato che, nonostante la ridondanza, con un numero di sinergie uguale a quello nel modello dei momenti, è possibile ottenere le stesse prestazioni.

Modello attuato da muscoli (capitolo 5) Questo modello, oltre ad essere ridondante (6 muscoli e 2 gradi di libertà) e ad avere vincoli biologici (forze positive), è caratterizzato dal fatto che la relazione tra attivazione muscolare e momenti ai giunti prodotti è fortemente non lineare. La relazione generale tra attivazione muscolare $\boldsymbol{M}$ e momenti ai giunti $\boldsymbol{\tau}$ è data da:

$$\boldsymbol{M} = \boldsymbol{R}(\boldsymbol{l}, \dot{\boldsymbol{l}})^{+}(\boldsymbol{\tau} - \boldsymbol{L}\boldsymbol{P}(\boldsymbol{l})) + \boldsymbol{Q}(\boldsymbol{l}, \dot{\boldsymbol{l}})\boldsymbol{z} \tag{6}$$

dove $\boldsymbol{R}(\boldsymbol{l}, \dot{\boldsymbol{l}})$ rappresenta la relazione non lineare tra attivazioni e momenti ai giunti, $\boldsymbol{R}(\boldsymbol{l}, \dot{\boldsymbol{l}})^+$ è la sua pseudo-inversa e $\boldsymbol{Q}(\boldsymbol{l}, \dot{\boldsymbol{l}})$ è una base ortogonale di $\ker(\boldsymbol{R}(\boldsymbol{l}, \dot{\boldsymbol{l}}))$. Analogamente a $\boldsymbol{w}$ nel modello delle forze, il termine $\boldsymbol{z}$ è un *nullshift* che può essere usato per variare il profilo dell'attivazione $\boldsymbol{M}$ senza modificare il momento ai giunti. Mostriamo che, per via del carattere non lineare di $\boldsymbol{R}(\boldsymbol{l}, \dot{\boldsymbol{l}})$, la dimensionalità delle attivazioni rispetto a quella dei momenti ai giunti esplode. Ciò risulta, seppur moderatamente, in un numero maggiore di sinergie necessarie al fine di ottenere le stesse prestazioni degli altri due modelli.

Conclusioni

Mostriamo che, nei limiti del modello sviluppato, il controllo tramite combinazione lineare di sinergie è fattibile. Mostriamo inoltre che la ridondanza nei sistemi muscolo-scheletrici potrebbe non influenzare il numero di sinergie necessario a controllarli e quindi che la complessità del controllo tramite sinergie muscolari potrebbe non dipendere dal numero di muscoli. Infine, mostriamo che le non linearità derivanti dalla dinamica del muscolo potrebbero invece aumentare la complessità del controllo.

Acknowledgements

It's difficult to get at this point where I am now, without the help of precious people.

Special thanks go to Cristiano Alessandro for his great supervision. With his amazing methodology and reasoning he followed me in all parts of this work, having always time for questions and constructive comments, and yes, also for fun. I am thankful to Juan Pablo Carbajal which gave me, with his stunning knowledge of mathematics and physics, precious insights and tips. I have to thank both Cristiano and Juan Pablo for the valuable support and the many suggestions during the composition of this manuscript and of the materials used to present my work.

I would like to thank Prof. Giuseppina Gini and Prof. Robert Riener, who gave me the possibility to do my work at the Sensory-Motor Systems Lab of the Swiss Federal Institute of Technology in Zürich.

I'm especially grateful to my parents, which always supported me all these years. With their education I learned to fight and strive for my objectives.

My deepest thanks, however, go to my wife Monica and to my children Jacopo and Iris. In these years, made of evenings and weekends spent on books, I have stolen a lot of time from my family. Despite all the difficulties and the sacrifices, they always supported me and believed in me, and gave me the energy needed to get here. Thank you!!

Finally, thanks to everybody I forgot to mention but who contributed, in any way, to this work and to my studies.

Robin

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Musculoskeletal systems, like our body, are very complex and are composed by a lot of components interacting together. We have a great quantity of bones and hundreds of muscles. Despite all this complexity, we can perform and learn a variety of tasks. We can stand, walk, run, jump, touch or take things, look behind and many, many other type of movements. We execute these tasks spontaneously and harmoniously without the need to think about it. Think for example of the simple, everyday action of taking a cup of coffee: in doing so we use dozens of degrees of freedoms and muscles, all present in our arm and hand, and we do not put, at least consciously, the minimum attention in this. How can we do that? How can our CNS (Central Nervous System) efficiently tackle all this complexity, while we, with our supercomputers and hi-tech materials, have still many problems in controlling much simpler systems?

Researches have tried to give an answer to these questions formulating several hypotheses. One of the them, is that the CNS simplifies motor control using a modular architecture, where the modules are called muscle synergies. Muscle synergies are patterns of motor commands that can be "linearly" combined in order to obtain the command patterns required to execute a task (i.e. reach a cup of coffee). This strategy would result in a dimensionality reduction of the control problem, and would therefore simplify control and learning.

Many authors have found experimental evidence of the existence of muscle synergies. The classical approach consists in analyzing EMG (Electromyographic) signals recorded from muscles, and try to extract components (i.e. synergies) which, combined, explain the variability in the dataset. Many questions, however, are still open. For example, the role of biomechanics in the synergy hypothesis is still not clear. We don't know if it's possible

to control a musculoskeletal system by linear combination of synergies, and we don't know how biomechanical features influence the hypothesis. In this thesis we try to give an answer to these questions using a computational approach. In particular we analyze the role of redundancy and nonlinearities in the synergy hypothesis and test if the control by means of muscle synergies is possible.

In the next sections we will first elaborate on the complexity in motor control, and will explain why muscle synergies may simplify it. Afterwards, we will explain and formalize the muscle synergy model. Then we review the state of the art on muscle synergies and finally we focus on the open problems, and on the objectives of this work.

1.1 Complexity in motor control

There is no doubt that the control of the musculoskeletal system is difficult. In the following we describe the factors that make it so complex, and explain conceptually how they are exhibited and what they imply from the control point of view.

Muscle redundancy In a musculoskeletal system like our body, there are more muscles than mechanical degrees of freedom, thus the system is redundant. In such a system the total coordinated action of many muscles leads to a certain movement. Since different muscles act on the same joint, muscles can be commanded in many different ways to solve a certain task, obtaining the same movement. As an example, we can reach a cup of coffee with a "relaxed" movement, or by co-contracting every muscle and increasing the stiffness of our arm. The forces exerted by the muscles are different, but we are still able to move along the same trajectory. Hence, in order to execute a task, our CNS has to "choose" among a number of infinite possible ways of actuating the involved muscles. In the rest of the thesis we will refer to muscle redundancy simply as *redundancy*. In section 2.2.2 we will formalize redundancy and explain the role it plays in our work.

Nonlinearities Musculoskeletal systems are nonlinear. This means that linear combinations of inputs do not lead to linear combinations of outputs. This implies that "similar" tasks may require quite different motor commands in order to be executed. As a result, the CNS could exploit past experience in a very limited way, and appropriate motor commands should be computed/learned almost from scratch for each new task.

Nonlinearities originate from almost each level of the musculoskeletal system. At the skeletal level (that can be described with a kinematic chain) there is a nonlinear relation between torque and joint movements. Muscles are nonlinear. They are controlled in activation by motor neurons, and produce a force which depends nonlinearly on their length and contraction velocity. In section 2.2.2 we will show the implications of nonlinearity on our work from a mathematical point of view.

Other factors Redundancy can be observed not only at the muscular level, but also at the neural level. Each muscle is composed by lot of muscle units, commanded by many motor neurons. The CNS can obtain the same muscle activation by recruiting different motor neurons and motor units. There is also redundancy at the kinematic level. The number of degrees of freedom is higher than the number of variables describing the task. In the human arm, for example, there are seven mechanical degrees of freedom without considering the hand. This means that a position of the hand in space (e.g. to reach a cup of coffee), can be obtained by different configurations of the shoulder, elbow and wrist joints. Tackling kinematic redundancy means deciding which particular configuration each joint has to take in order to reach a desired configuration. Another factor that makes control difficult are delays between the controller (i.e. CNS) and actuators (i.e. muscles). Similarly there are delays between sensory informations, originating for example from muscle spindles, and the controller.

Although the musculoskeletal system is nonlinear and redundant, many studies have shown that different subjects employ similar kinematics and muscle activation patterns to solve a given task. How does the CNS choose this solution among the infinite number of motor commands that solve a task? This problem was formulated for the first time by Nikolai Bernstein [11], and it is often referred as *degrees of freedom problem*. He suggested that muscles are not controlled individually, rather the CNS relays on a modular architecture where the modules are combined in order to obtain a complex behavior. This would reduce the number of variables to be controlled, thus simplifying motor control and learning. The Bernstein problem was studied and implemented in different ways, one of them is the model of muscle synergies.

1.2 Muscle synergy hypothesis

In order to execute a task (e.g. reach a cup of coffee) the CNS has to generate appropriate motor commands (e.g. muscle activation) (Figure 1.1). How does the CNS compute the required muscle activations?

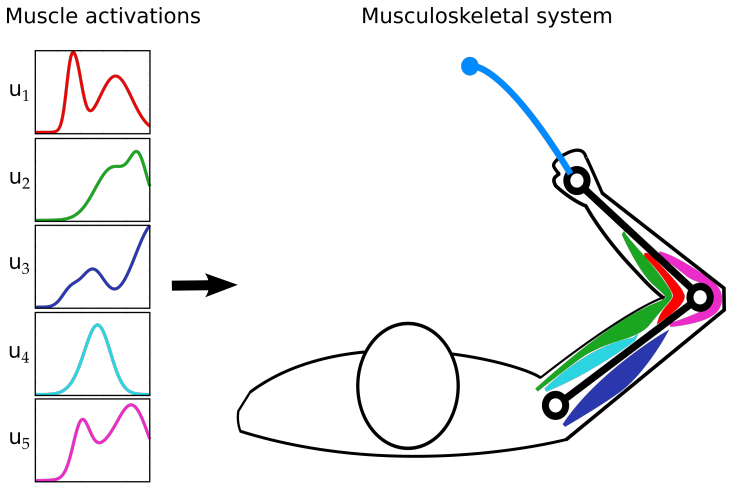


Figure 1.1: In order to execute a task (e.g. light blue reaching task) the CNS has to generate the muscle activations $u_i(t)$ which lead to the desired movement.

According to the muscle synergy hypothesis, motor commands are generated by combining predefined "modules", called muscle synergies. These are defined as specific co-activation patterns among groups of muscles. Hypothetically, the CNS combines few synergies in order to obtain complex behaviors and solve specific tasks. This leads to a dimensionality reduction of the control problem, as the CNS only has to choose how to combine the synergies, instead of calculating each individual muscle activation. As an example, think to a painter creating its whole color palette by only mixing different quantities of few colors. We can think to the color palette as the set of complex tasks to be executed, and to the few color generators as the set of synergies.

One of the most important aspects of the muscles synergy hypothesis is that synergies are invariant across tasks (Figure 1.2). As an example, authors have found that in frogs some synergies are shared across kicking, jumping, walking etc. [24]. There is also some evidence that synergies may be invariant across individuals, that is, subjects from the same species may share similar synergies [23, 30, 25].

From a mathematical point of view two main models of muscle synergies have been proposed (see [2] for a more extensive review):

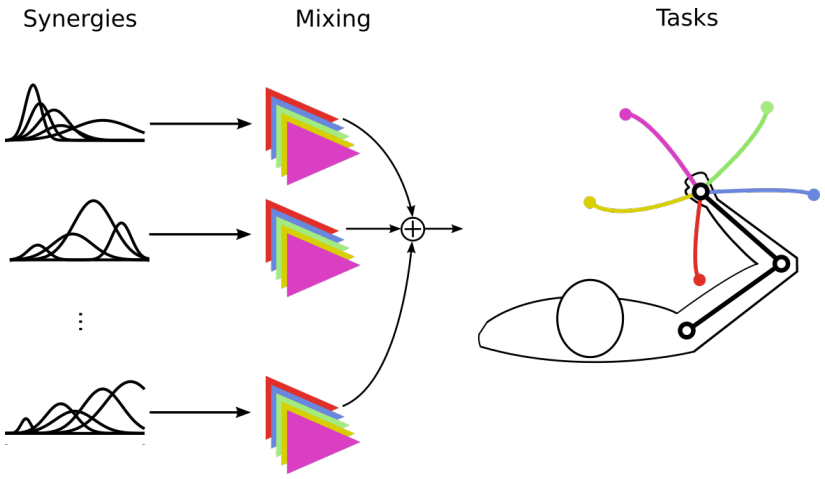


Figure 1.2: A small set of synergies (left) are mixed in different ways (center) in order to obtain muscle activations required to execute different tasks (right).

- Time invariant synergies
- Time-varying synergies

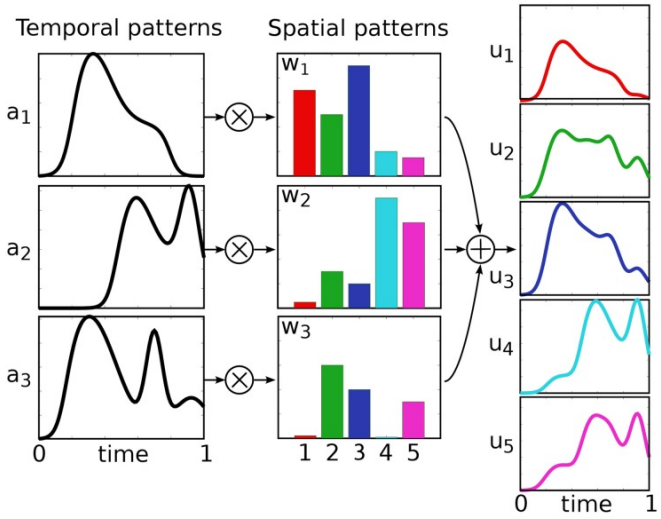
In the following we will describe and formalize the two models.

Time invariant synergies

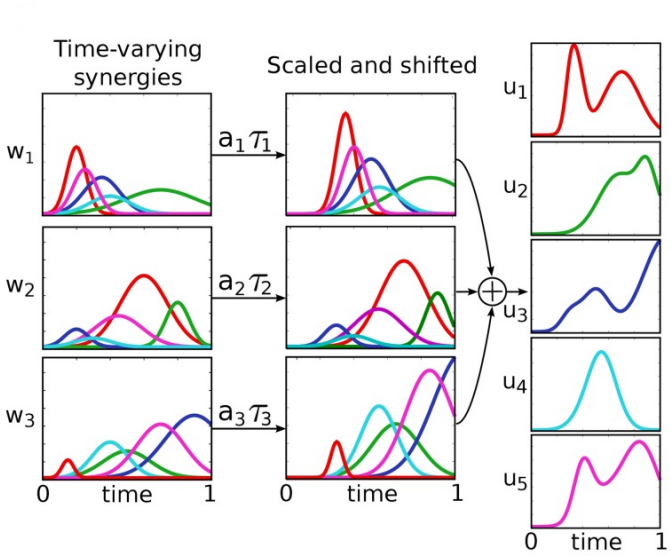
Time invariant synergies are also called *synchronous synergies*, *spatially fixed synergies* or *muscle modes*. Each synergy $\mathbf{w}_j$ corresponds to a spatially fixed pattern of muscle activations. Temporal structure is instead specified by the 1-dimensional real functions of time $a_j(t)$ that serve as synergy mixing coefficients (Figure 1.3a). Formally, the total activation $\boldsymbol{u}(t)$ given as input to muscles is:

$$\boldsymbol{u}(t) = \sum_{j=1}^{n} a_j(t)\mathbf{w}_j \tag{1.1}$$

where n is the number of synergies. The vectors $\mathbf{w}_j$ represent, therefore, the task-independent synergies which are combined linearly by means of the task-dependent and time-dependent functions $a_j(t)$. Given that the elements $a_j(t)$ are 1-dimensional real functions, this model results in a dimensionality reduction only if the number of synergies is lower than the number of muscles (see [2] for more details).



(a)



(b)

Figure 1.3: Models of muscle synergies. (a) Time invariant synergies model: the spatially fixed synergies $\mathbf{w}_j$ are modulated by the temporal patterns $a_j(t)$ and summed together to obtain the complex activations $\boldsymbol{u}(t)$ which solve a task. (b) Time-varying synergies model: the time-varying synergies $\mathbf{w}_j(t)$ are scaled by the coefficients a_j, and shifted in time by the onset time τ_j. Image adapted with permission from [2].

Time-varying synergies

In this model, synergies $\mathbf{w}_j(t)$ are time-varying patterns of muscles activations that can be shifted in time and modulated in amplitude by the scalar

coefficients τ_j and a_j (Figure 1.3b). Formally, the total muscle activation $\boldsymbol{u}(t)$ is given by:

$$\boldsymbol{u}(t) = \sum_{j=1}^{n} a_j \mathbf{w}_j(t - \tau_j) \tag{1.2}$$

where n is the number of synergies. The vector-valued functions $\mathbf{w}_j(t)$ represent the task-independent synergies, which are combined by means of the task-dependent coefficients a_j, and are time-shifted by the task-dependent onset times τ_j. Since these variables are scalars, the input-space is reduced to a $2 \cdot n$ dimensional space, with n the number of synergies (see [2] for more details).

In this work we implement synergies according to the time-varying model without considering the shifting coefficients τ_j (i.e. $\tau_j = 0 \ \forall j$). Note, however, that for any synergy $\mathbf{w}_j$ shifted by an onset time τ_j, we can define another synergy $\mathbf{w}_k$ as follows:

$$\mathbf{w}_k(t) = \mathbf{w}_j(t - \tau_j) \tag{1.3}$$

Therefore, except for the number of synergies, our model is equivalent to the time-varying model.

1.3 State of the art

In this section we review the state of the art on the muscle synergies hypothesis.

The classical approach to investigate the muscle synergy hypothesis consists in recording EMG (Electromyographic) signals from a set of muscles during the execution of a task (e.g. reaching task). Decomposition algorithms are then applied in order to extract components (i.e. synergies) that can reconstruct the dataset. If these components are invariant across different experimental conditions (e.g. reaching targets, movement speed etc.), they are considered an indirect evidence of the existence of an underlying modular architecture. In order to extract components, researchers use algorithms like PCA (Principal Component Analysis), ICA (Independent Component Analysis), NMF (Non-Negative Matrix Factorization) and other. The quality of the extracted synergies is then evaluated using statistical measures (e.g. R^2) that assess how much data variability can be explained by linear combinations of the extracted components. The choice of the algorithm depends on assumptions made a-priori, and influences the obtained results [61]. Furthermore, the number of synergies is mostly defined

a-priori [2]. Despite these limitations, this approach has provided insights on the hypothesized modular control strategy both in humans and in animals.

In [23] the authors were able to reconstruct the EMG dataset recorded from human reaching movements with 4-5 time-varying synergies. Similar results are shown in [21] where a few number of synergies, robust across changes in reaching direction and speed, were found. In the context of reaching tasks, it was found that muscle synergies can account for both movements and forces in multiple directions [8]. In [20] two time-varying synergies captured most of muscle pattern variability observed during ball catching exercises. In particular, the authors observed that one of the synergies was activated always short before catching the ball, suggesting that synergy recruiting is modulated by visual informations. Similar experiments have been performed by other authors reporting experiments on muscle synergies extracted from the arm and/or hand of humans [7, 19] and primates [54]. In [13] the muscles synergy hypothesis is supported by the result that a minimum effort recruiting strategy of muscle synergies explains the recorded muscle activations better than a minimum effort recruiting strategy of individual muscles. The EMG signals were recorded during the generation of isometric forces at the hand.

Others have done similar experiments on other type of tasks. In [42] a small set of synergies explained the activations recorded during cycling exercises and were stable across different experimental conditions. In [43] five synergies account for human locomotion in a variety of conditions. Similar results are obtained in [14] during running. Muscle synergies have been extracted also during postural balance [63]. In this context, several authors have found common synergies between postural control stepping and non-stepping responses [18], balance and walking [17], cycling and walking [25].

Many experiments have been done also on animals (frogs, cats, primates). In [24], for example, three synergies explained the muscle patterns recorded during frog kicking and were found to be stable across individuals and partially across behaviors (walking, jumping and swimming).

The debate on what synergies represent is still open. Modular structures, like muscle synergies, have in fact been observed at different levels. Some authors have found that muscle synergies may be expressions of neurons organized in spinal primitives [38]. Other analyzed them from a task-space perspective, for example extracting kinematic synergies which reconstruct joint kinematic trajectories [66, 57, 29, 60, 73] or at the force level, finding that the force obtained by co-stimulation of two spinal cord sites in the frog, was very similar to the vector summation of the forces obtain by separately stimulating the two sites [51]. These different representations, however,

could be the results of a unique modular organization of the motor control. In [60], for example, the authors have recorded EMG signals during human reaching and grasping exercises. Simultaneously they measured kinematic variables, and found that synergies extracted from the EMG datasets are at the origin of the kinematic synergies.

There are also experiments which discredit the muscle synergy hypothesis. For example in [48] the authors demonstrate, through cadaveric experiments, that muscle synergies may emerge simply from biomechanical couplings (i.e. interaction between joints, bones, muscles etc.), without the need to conclude that they are the results of some neural modular architecture.

In general, the muscle synergy hypothesis is very difficult to verify or falsify. One of the main difficulties with the approach based on the analysis of EMG dataset is that there is no direct task verification. That is, ones the synergies are extracted it is not possible to directly verify if they solve the original task. Some authors therefore tried to follow other approaches. In [9] the authors recorded EMG signals from reaching tasks in a virtual PC environment. Reachings were executed by subjects exerting forces on a handle. They extracted a set of synergies from the EMG dataset, and calculated the mapping between the dataset reconstructions and the forces applied to the handle. Then they applied transformations (i.e. virtual surgeries) to the mapping between forces on the handle and movements of the cursor on the PC. Virtual surgeries were of two types: compatible with the synergies (i.e the forces needed to execute the tasks could still be generated by a suitable combination of the extracted synergies), and incompatible with the synergies. In the case of compatible surgery the subjects were still able, after a short adaption phase, to execute the tasks. In the case of incompatible surgeries instead, the subjects were no more able to execute the tasks. The idea was that if control was not based on some modular architecture, the subjects would be equally able to learn the tasks in both cases. A very similar approach has been followed in [34], where researchers have found that the extracted synergies are robust to visuomotor rotations.

Since musculoskeletal systems are nonlinear, synergies extracted from EMG datasets may not lead to the expected task performance. Furthermore, muscles that are not recorded during experiments can potentially play an important role for the movements analyzed [3]. Some researches, therefore, have considered task performance variables in their experiments. In the model of "functional synergies", components are extracted from a dataset of both EMG and task variables. This has allowed researchers to find a correspondence between muscle synergies and task properties [18, 62]. Another

model that is grounded on task variables is that of the "spinal force fields". It originated from the observation that the forces generated through spinal stimulation of the frog limb, depend on the posture of the limb ([35, 51], see [3] for a more extensive review on functional synergies and spinal force fields). Another approach has been followed in [27] and [28] where a possible relation between synergy recruiting parameters and task discriminating properties is exploited during the synergy extraction. The method results in the identification of the minimum number of muscle synergies which can explain most of the variability in the task space.

Although many researchers are already moving from a perspective considering only the input-space (i.e. synergy activation pattern), to one where the task-space perspective (i.e. task execution) becomes more and more important, the musculoskeletal dynamics is often not taken into account, and many questions regarding the relation between synergies and task execution remain open. We don't know which role is played by the musculoskeletal dynamics, or how its properties influence the results. We don't even know if a small set of primitives may actually be able to control a complex musculoskeletal system. One possible strategy to tackle these questions is to follow a computational approach, namely to define a dynamical system (e.g. model of limb) and a set of tasks (e.g. reaching tasks), and see if it's possible to find a set of synergies that are able to solve the tasks. In [53], for example, a mathematical formulation which leads to the synthesis of a set of primitives in accordance with the spinal force fields model is proposed. The synthesized primitives are able to control a two-dof planar kinematic chain, and the authors prove that the dimensionality of the task space sets a lower bound to the number of primitives needed. A recent study attaining the same conclusion is [26], where the authors show that the number of synergies required for a generic biomechanical system increases with the dimensionality of the task-space. In [10], a model of the frog hindlimb is first reduced to a low-dimensional model. Then an optimization process synthesizes synergies on this reduced model from a representative set of desired actuations. The synthesized synergies were similar to those found experimentally by other researches. In [58] a simulated musculoskeletal model of human lower limb is driven by five primitives extracted from EMG signals of two subjects. Finally in [1, 2] a method is proposed for the generation of open loop controllers based on synergies. The method, called DRD (Dynamic Response Decomposition), explicitly integrates the musculoskeletal dynamics in the procedure used to synthesize synergies, and exploits the relation between input space and task space by assuming that muscle synergies are solutions to representative tasks. In this thesis we extend this

work trying to understand how some important biomechanical features of musculoskeletal systems influence the synergy hypothesis.

1.4 Objectives

In the previous section we claimed that the approach based on the extraction of input primitives from recorded signals (e.g. EMG, kinematics trajectories etc.) represents only an indirect evidence of the existence of muscle synergies, even if task variables are integrated in the analysis. The musculoskeletal dynamics, which represents the relation between input-space and task-space is rarely explicitly considered. Therefore, it is not clear if muscle synergies represent a feasible approach to control the musculoskeletal system. Some researchers have shown computationally [53, 10, 2] that it is possible to control a dynamical system by means of muscle synergies. However a systematic analysis on how biomechanical features influence the synergy model is still missing. In this thesis we investigate (1) the extent to which *redundancy* and *nonlinearities*, two important features of musculoskeletal systems, influence the muscle synergy hypothesis, and (2) whether a musculoskeletal model that features these properties can be controlled by linear combinations of synergies.

In order to answer these questions we synthesize synergies for systematically more complex models of the human arm. We start with a 2-joint kinematic chain actuated in torque space, and we validate the results obtain in [2]. Then we actuate the kinematic chain by means of 6 forces, thus leading to a *redundant* model. Finally we generate the 6 forces by means of *nonlinear* muscles controlled in muscle activation.

Synergies are synthesized using the DRD method for each of these models, and performance are evaluated for the execution of reaching tasks. The synergies will be formulated according to the time-varying synergy model, without onset time[1]. Furthermore we will assume, according to [2], that synergies represent solutions to specific task instances. That is, the synergies which are used to solve reaching tasks, are solutions to "specific" reaching tasks themselves.

[1] We have shown in section 1.2 that our model is equivalent to the time-varying synergy model, except for the resulting number of synergies.

1.5 Structure of the thesis

The first part of this thesis is the introduction (chapter 1), which starts with a brief initial explanation on the issues faced in this work. We then elaborate on the complexity of motor control in musculoskeletal systems and introduce the synergy hypothesis followed by a review on the state of the art. Finally we describe the objectives of this work.

In chapter 2 we give the mathematical details of the DRD method (section 2.1), and of the musculoskeletal models used (sections 2.2.3, 2.2.4 and 2.2.5). The chapter terminates with a section regarding the procedures used in our experiments (section 2.3).

In chapter 3 we show the results obtained on the model actuated by means of torques. In chapters 4 and 5 we show, respectively, the results obtained on the redundant model actuated by means of forces and the results obtained on the model actuated by means of nonlinear muscles.

In chapter 6 we conclude our work with a summary of the results obtained and a discussion on future works.

In the appendices A and B we give details about the procedure used to lump the musculoskeletal model. Finally, in appendix C, we show a procedure implemented for this work to include dynamical system boundaries into the DRD method.

Chapter 2

Methods

In this chapter we introduce the mathematical models and tools used in our work. First we introduce DRD, a general method to generate inputs to a dynamical system as linear combinations of primitives, used here to represent and synthesize synergies. Then, in section 2.2, we describe the musculoskeletal model of human upper limb developed for our experiments and provide the details of the torque, force and activation actuated kinematic chain built on top of it. Finally, in section 2.3, we provide the relevant data and procedures needed to execute the experiments.

2.1 The DRD method

The *Dynamics Response Decomposition (DRD)* is a general method to generate inputs for a given dynamical system as linear combinations of primitives [15, 1, 2]. We used the DRD framework to perform our analysis. In what follows we describe the method, and unfold its mathematical details. To support the explanation we use a generic multi-joint kinematic chain as exemplary dynamical system, however DRD is a general method applicable to a variety of dynamical systems.

2.1.1 Definitions

Consider a generic dynamical system described by the differential operator $\mathcal{D}$

$$\mathcal{D}\left(\boldsymbol{q}(t)\right) = \boldsymbol{u}(t) \tag{2.1}$$

where $\boldsymbol{q}(t) \in \mathbb{R}^{n_q}$ is a vector of n_q configuration variables and $\boldsymbol{u}(t) \in \mathbb{R}^{n_u}$ is the actuation vector at time t. In our example $\mathcal{D}$ represents the equation of

motion of a kinematic chain, $\boldsymbol{u}(t)$ is the vector of the torques applied to the joints, and $\boldsymbol{q}(t)$ represents the joint positions.

Inspired by the muscles synergy hypothesis, the actuation $\boldsymbol{u}(t)$ is represented as linear combination of time-dependent primitives $\{\boldsymbol{\phi}_i(t)\}$ (i.e. synergies) with mixing coefficients $\{b_i\}$:

$$\boldsymbol{u}(t) = \sum_{i=1}^{n_\phi} \boldsymbol{\phi}_i(t) b_i \tag{2.2}$$

If time is discretized we can represent the actuation over n_t time samples as a $n_t \cdot n_u \times 1$ vector $\boldsymbol{u}$:

$$\boldsymbol{u} = \sum_{i=1}^{n_\phi} \boldsymbol{\phi}_i b_i = \boldsymbol{\Phi b} \tag{2.3}$$

where $\boldsymbol{\Phi} = [\boldsymbol{\phi}_1 \dots \boldsymbol{\phi}_{n_\phi}] \in \mathbb{R}^{n_t \cdot n_u \times n_\phi}$ is the matrix that contains each discretized primitive in its columns and $\boldsymbol{b} = [b_1 \dots b_{n_\phi}]^T$.

The primitives $\{\boldsymbol{\phi}_i(t)\}$ are themselves actuations, and when applied to the dynamical system individually they produce the trajectories $\{\boldsymbol{\theta}_i(t)\}$ in configuration space, called *dynamic responses (DR)*:

$$\mathcal{D}\left(\boldsymbol{\theta}_i(t)\right) = \boldsymbol{\phi}_i(t). \tag{2.4}$$

If time is discretized we can represent the dynamic response $\boldsymbol{\theta}_i(t)$ with a $n_t \cdot n_q \times 1$ vector $\boldsymbol{\theta}_i$ and the whole set with the $n_t \cdot n_q \times n_\theta$ matrix $\boldsymbol{\Theta} = [\boldsymbol{\theta}_1 \dots \boldsymbol{\theta}_{n_\theta}]$.

For our kinematic chain $\boldsymbol{\theta}_i(t)$ represents the joints position, while $\dot{\boldsymbol{\theta}}_i(t)$ and $\ddot{\boldsymbol{\theta}}_i(t)$ their velocity and acceleration respectively at time t.

2.1.2 Dynamic response decomposition

The DRD generates the inputs to a dynamical system that solve a desired task, and it approximates such inputs as linear combination of primitives. A task is defined by constraints in the phase space. For example a task could be defined as moving the joints of our kinematic chain from an initial (at time $t = t_0$) to a final (at time $t = t_f$) position with some initial and final velocities. This task can be formulated with the following constraints:

$$\begin{aligned} \boldsymbol{q}(t_0) &= \boldsymbol{q}_0, \quad \dot{\boldsymbol{q}}(t_0) = \dot{\boldsymbol{q}}_0 \\ \boldsymbol{q}(t_f) &= \boldsymbol{q}_f, \quad \dot{\boldsymbol{q}}(t_f) = \dot{\boldsymbol{q}}_f \end{aligned} \tag{2.5}$$

We want to actuate the dynamical system in order to satisfy the task constraints.

The DRD method first finds a trajectory $\boldsymbol{q}(t)$ satisfying the constraints, and then computes the corresponding actuation $\boldsymbol{u}(t)$ leading to the found trajectory. Given the set of DRs associated to the dynamical system, a trajectory $\boldsymbol{q}(t)$ is found by interpolation, searching for a linear combination of DRs that fulfills the task constraints:

$$\boldsymbol{q}(t) = \sum_{i=1}^{n_\theta} \boldsymbol{\theta}_i(t) a_i \tag{2.6}$$

where $\boldsymbol{\theta}_i(t)$ are the dynamic responses and a_i the mixing coefficients. If time is discretized we can write Eq. (2.6) in matrix form as:

$$\boldsymbol{q} = \boldsymbol{\Theta}\boldsymbol{a} \tag{2.7}$$

where $\boldsymbol{\Theta}$ is the matrix containing the DRs in its columns and $\boldsymbol{a} = [a_1 \dots a_{n_\theta}]^T$ is the vector of mixing coefficients. See [1] for an evaluation of the quality of the DRs used as interpolators for the trajectory of reaching tasks on a planar kinematic chain.

The mixing coefficients $\boldsymbol{a}$ can be found by solving the following linear system of equations:

$$\begin{pmatrix} \boldsymbol{\theta}_1(t_0) & \dots & \boldsymbol{\theta}_{n_\theta}(t_0) \\ \boldsymbol{\theta}_1(t_f) & \dots & \boldsymbol{\theta}_{n_\theta}(t_f) \\ \dot{\boldsymbol{\theta}}_1(t_0) & \dots & \dot{\boldsymbol{\theta}}_{n_\theta}(t_0) \\ \dot{\boldsymbol{\theta}}_1(t_f) & \dots & \dot{\boldsymbol{\theta}}_{n_\theta}(t_f) \end{pmatrix} \boldsymbol{a} = \boldsymbol{M}\boldsymbol{a} = \begin{pmatrix} \boldsymbol{q}_0 \\ \boldsymbol{q}_f \\ \dot{\boldsymbol{q}}_0 \\ \dot{\boldsymbol{q}}_f \end{pmatrix} = \boldsymbol{P} \tag{2.8}$$

where $\boldsymbol{P}$ is the vector of task constraints and $\boldsymbol{M}$ is the so called *alternant matrix*, containing the DR evaluated at the timestamps where the task constraints are defined (t_0 and t_f in the example).

2.1.3 Solution composition

Once a trajectory has been found, the corresponding actuation $\tilde{\boldsymbol{u}}(t)$ can be obtained by applying the differential operator $\mathcal{D}$ (inverse dynamics):

$$\mathcal{D}\left(\boldsymbol{q}(t)\right) = \mathcal{D}\left(\boldsymbol{\Theta}(t)\boldsymbol{a}\right) = \tilde{\boldsymbol{u}}(t) \tag{2.9}$$

In our example $\tilde{\boldsymbol{u}}(t)$ represents the torques required to obtain a joint trajectory that fulfills the constraints.

Finally DRD approximates this actuation as a linear combination of primitives (see Eq. (2.3)). In general this consists in solving the following minimization:

$$\boldsymbol{b} = \underset{\boldsymbol{b}}{\operatorname{argmin}} \, ||\tilde{\boldsymbol{u}}(t) - \boldsymbol{\Phi}(t)\boldsymbol{b}|| \tag{2.10}$$

where $||\cdot||$ is the Euclidean norm. If time is discretized this problem can be solved using the Moore-Penrose pseudoinverse of $\boldsymbol{\Phi}$

$$\boldsymbol{b} = \boldsymbol{\Phi}^{+}\tilde{\boldsymbol{u}} \tag{2.11}$$

If the desired actuation $\tilde{\boldsymbol{u}}(t)$ is not in the linear span of the primitives, the obtained actuation $\boldsymbol{u}(t) = \boldsymbol{\Phi}(t)\boldsymbol{b}$ will differ from $\tilde{\boldsymbol{u}}(t)$ by an error $\boldsymbol{\varepsilon}_u(t) = \tilde{\boldsymbol{u}}(t) - \boldsymbol{u}(t)$.

2.1.4 Performance measures

To evaluate the performance of DRD, we define three measures:

- the *interpolation error* measures the ability of the DRs $\boldsymbol{\Theta}$ to generate a trajectory satisfying the task constraints $\boldsymbol{P}$;
- the *projection error* measures the ability of the primitives $\boldsymbol{\Phi}(t)$ to approximate the actuation $\tilde{\boldsymbol{u}}(t)$;
- the *forward dynamics error* measures the error between the task constraints $\boldsymbol{P}$ and the trajectory obtained when the dynamical system is actuated by $\boldsymbol{u}(t) = \boldsymbol{\Phi}(t)\boldsymbol{b}$ (which may be different from $\tilde{\boldsymbol{u}}(t)$).

Interpolation error The interpolation error measures the distance between the interpolated trajectory $\boldsymbol{q}(t) = \boldsymbol{\Theta}(t)\boldsymbol{a}$ and the task constraints. The distance measure is the Euclidean norm. Given k task constraints the interpolation error err_I is defined as:

$$err_I = \sqrt{\sum_{k \in K} err_{I_k}^2} = \sqrt{\sum_{k \in K} \|\boldsymbol{q}_k - \boldsymbol{\theta}_k \boldsymbol{a}\|^2} \tag{2.12}$$

where err_{I_k} is the interpolation error associated to the k-th task constraint, $\boldsymbol{q}_k$ is the desired value, and $\boldsymbol{\theta}_k$ the DRs evaluated at the corresponding timestamp. If the linear system of equations (2.8) has exact solution then the interpolation error is zero, meaning that the interpolated trajectory satisfies all the task constraints exactly.

Projection error The projection error measures the distance between the actuation $\tilde{\boldsymbol{u}}(t)$ which solves the tasks and the one found as linear combination of primitives:

$$err_P = \sqrt{\int_T \|\tilde{\boldsymbol{u}}(t) - \boldsymbol{\Phi}(t)\boldsymbol{b}\|^2 \mathrm{d}t} = \sqrt{\int_T \|\boldsymbol{\varepsilon}_u(t)\|^2 \mathrm{d}t} \tag{2.13}$$

This error is zero only if the desired actuation is in the linear span of the primitives.

The projection error has the same unit of measure of the actuation $\tilde{\boldsymbol{u}}(t)$, consequently it is not possible to numerically compare errors from different type of actuations (e.g torques and forces). We therefore define the *relative projection error* as the adimensional quantity:

$$err_{P_r} = \frac{err_P}{\|\tilde{\boldsymbol{u}}(t)\|} \tag{2.14}$$

which can be used to compare errors from actuations with different unit of measures. In this work we will use both the projection error and the relative projection error.

The idea behind the projection error is that if the actuation $\tilde{\boldsymbol{u}}(t)$ leads to a trajectory satisfying the task constraints, then another actuation $\boldsymbol{u}(t) \neq \tilde{\boldsymbol{u}}(t)$ might most probably not satisfy them and the error in executing the task will be related to how much they are different. This assumption does not hold in general for at least two reasons:

- for a redundant system there may be multiple actuations that lead to the same trajectory. However, these solutions will have different projection error;
- there may be various trajectories satisfying the task constraints. Each trajectory, however, would be obtained actuating the dynamical system with a different actuation. These actuations would have different projection errors.

Forward dynamics error The forward dynamics error measures the distance between the trajectory $\boldsymbol{q}(t)$ obtained by actuating the dynamical system with $\boldsymbol{\Phi}(t)\boldsymbol{b}$ and the task constraints. The distance measure is as usual the Euclidean norm:

$$err_F = \sqrt{\sum_{k \in K} err_{F_k}^2} = \sqrt{\sum_{k \in K} \|\boldsymbol{q}_k - \boldsymbol{q}(t_k)\|^2} \tag{2.15}$$

where err_{F_k} is the forward dynamics error associated to the k-th task constraint, $\boldsymbol{q}_k$ is the constraint value and $\boldsymbol{q}(t_k)$ the obtained trajectory evaluated at the corresponding timestamp. The forward dynamics error is therefore a direct measure of task performance as it shows how much the real trajectory differs from the constraints when the system is actuated with the linear combination of primitives.

2.2 The musculoskeletal model

In this section we provide the details of the musculoskeletal model used in our work. Experiments involving the synergy hypothesis have been done in the past mainly on limb segments of animals and humans, although many researches also considered other types of synergies, for example those supposed to be involved in postural balance. In this work, we used a model of upper limb because it is simple but at the same time allows to capture the essential properties of movement, and to address redundancy and nonlinearities affecting musculoskeletal systems. Defining the particular characteristics of the model however is not trivial. Many researches have faced this issue and have proposed their own solutions based on their objectives. In [46] the authors use a two-joints kinematic chain with six Voigt muscles. In [44] a very similar model is proposed while in [49, 52, 45] Hill muscles are used. In [39] a simple but effective three-links limb model is actuated with Voigt muscles. Similar models with more complex limb or muscle configurations can be found in many other works [47, 55, 6, 50, 36]. Other authors have build very detailed three-dimensional musculoskeletal models of the upper limb with all bones, lots of muscles and degrees of freedom using available data from literature [41] or medical images [32, 33].

In this work we model the human upper limb with a two-joint kinematic chain. In the following sections we give the details on the skeletal model and define different ways of actuating it.

2.2.1 Skeletal model

The human arm is considered to have seven degrees of freedom: three in the shoulder, two in elbow, and two in the wrist. For simplicity we consider the wrist rigidly attached to the elbow, and we restrict our analysis to the horizontal plane, without considering the gravity. Furthermore we consider homogeneous mass distribution. We end up with a planar kinematic chain with two segments representing the upper arm and the forearm respectively, and two rotational joints representing the glenohumeral and elbow joints (Figure 2.1). In this manuscript we will refer to these joints as *shoulder* joint and *elbow* joint respectively. The shoulder joint is attached to a rigid body representing the scapula-clavicle complex. The joint axes are perpendicular to the horizontal plane.

Anthropometric data are taken from [52] and are reported in Table 2.1[1].

[1] In [52] the moment of inertia is given with respect to the joints, while for our purpose we needed the moment of inertia with respect to the center of masses of the two segments.

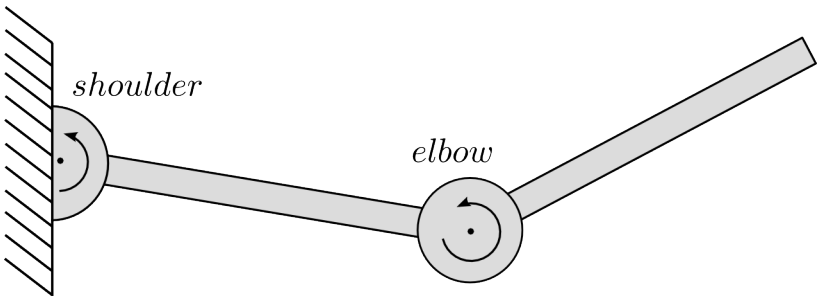


Figure 2.1: Kinematic chain representing the skeletal of a human arm model with the elbow and shoulder joint and the forearm and upper arm segments.

Table 2.1: Anthropometric data of the skeletal model.

	Forearm	Upper arm	Unit
Mass	1.43	1.82	Kg
Length	0.333	0.309	m
Center of Mass[1]	0.165	0.135	m
Moment of Inertia[2]	0.0186	0.0178	$Kg \cdot m^2$

[1] Distance from proximal joint
[2] Relative to the Center of Mass

2.2.2 Actuation

In order to investigate the impact of redundancy and nonlinearities on the synergy hypothesis, we systematically added these two biomechanical features to our kinematic chain. In this way it was possible to analyze their contributions separately. Depending on how the kinematic chain is actuated we define three models:

1. *Torque Model*: the kinematic chain is actuated by means of torques applied to the joints. This model is not redundant as the number of inputs is equal to the number of joints;

2. *Force Model*: the kinematic chain is actuated by means of 6 forces applied to the links. This model is redundant as the number of forces exceeds the number of joints;

We have recalculated them by applying the Huygens-Steiner theorem, best known as parallel axis theorem. The obtain data is comparable to that given by other authors [44, 49, 65].

3. *Activation Model*: in this model the 6 forces applied to the links are generated by means of nonlinear muscle models, which receive muscle activations as inputs.

Before looking at the details of the models above, we shortly define the concept of Redundancy and Nonlinearity.

Redundancy

In a human arm there are 7 degrees of freedom (dof) without considering the hand. These are controlled by dozens of muscles thus, the same joint configuration can be obtained by different muscle contraction patterns. A long standing question in neuromotor control is how the CNS "chooses" among infinite number of muscle activation patterns that solve a task and how certain patterns are chosen instead of others. This is the so called *degrees of freedom problem*, formulated for the first time by Bernstein [11]. In this section we define the concept of redundancy and explain the role it plays in our work.

In the first instance, a mechanical system is said to be redundant when it has more inputs than degrees of freedom (dof). As a result a theoretically infinite number of different inputs can lead to the same output. From the mathematical point of view, for a redundant dynamical system represented by the differential operator $\mathcal{D}$, the inverse dynamics problem of finding the unknown input $\boldsymbol{u}(t)$ leading to the known configuration $\boldsymbol{q}(t)$, is underdetermined, that is:

$$\exists \boldsymbol{u}_1(t) \neq \boldsymbol{u}_2(t) \quad s.t. \quad \boldsymbol{q}(t) = \mathcal{D}^{-1}\left(\boldsymbol{u}_1(t)\right) = \mathcal{D}^{-1}\left(\boldsymbol{u}_2(t)\right) \tag{2.16}$$

Finding a solution to the inverse dynamics in this case may be non-trivial, even if the input is chosen with some criteria. We will show that unfortunately this is exactly the case in this work.

How does redundancy apply to our musculoskeletal model? For simplicity let's first consider a single rotational joint actuated by a torque producing a certain angular acceleration. In this case the input $\boldsymbol{u}(t)$ to the dynamical system is the torque τ while the configuration $\boldsymbol{q}(t)$ is characterized by the joint angular position θ and velocity $\dot{\theta}$. The differential equation characterizing the system is:

$$\tau = I\ddot{\theta} \tag{2.17}$$

where I is the joint moment of inertia. This relation is one-to-one and, given an initial configuration, it is possible to determine the applied torque from the joint kinematics and vice-versa. Now consider another dynamical system

characterized by a joint actuated by means of two unconstrained forces f_1 and f_2 acting on the joint through two lever arms r_1 and r_2. The differential equation characterizing the system is:

$$r_1 \times f_1 + r_2 \times f_2 = I\ddot{\theta} \tag{2.18}$$

Once the two forces are known, the angular acceleration $\ddot{\theta}$ is uniquely determined. The contrary instead cannot be said. Given the angular acceleration $\ddot{\theta}$ (i.e. the joint kinematics) there are infinite combinations of f_1 and f_2 which generate it.

The discussion above generalizes to our musculoskeletal model. We have 2 degrees of freedom, namely the shoulder and the elbow joints. When we actuate the model with two torques, one per joint, the relation between actuation and kinematics is one-to-one and there is no redundancy. If instead we actuate the kinematic chain by means of forces, their total action produces a certain torque but the inverse dynamics problem of finding the contribution of each force given the joint kinematics is underdetermined and has infinite solutions.

Nonlinearity

In this section we explain the concept of nonlinearity and how its comes to play in our work.

In a nonlinear dynamical system the output is related to the input by a nonlinear relation. For such a system the superposition principle is not valid, meaning that in general it is not possible to linearly decompose the output in order to deduce a possible input decomposition. Similarly, if we know the input required to obtain a desired output, we cannot use this information to generate other inputs leading to other desired outputs. On the other hand, this is exactly what the synergy hypothesis speculates, that there are a set of input primitives (i.e. synergies) which linearly combined generate inputs leading to the execution of a desired task.

As described at the beginning of this section, we actuate the musculoskeletal model by means of torques applied to the joints, by means of forces, and by means of muscle activations. For a generic kinematic chain, the relation between torques applied to the joints and the kinematics is already not linear. We are however interested in the non-linearity which affects the relation between the input to the kinematic chain and the torque actually generated on the joints. Let's represents our musculoskeletal model actuated by torques with the usual differential operator $\mathcal{D}$:

$$\mathcal{D}(\boldsymbol{q}(t)) = \boldsymbol{\tau}(t) \tag{2.19}$$

where $\boldsymbol{q}(t)$ is the configuration and $\boldsymbol{\tau}(t)$ the actuation torque. Let's say that to execute a desired trajectory $\tilde{\boldsymbol{q}}(t)$ we have found a linear decomposition of the desired torque $\tilde{\boldsymbol{\tau}}(t)$ as follows:

$$\mathcal{D}(\tilde{\boldsymbol{q}}(t)) = \tilde{\boldsymbol{\tau}}(t) = a_1\boldsymbol{\tau}_1(t) + a_2\boldsymbol{\tau}_2(t) \tag{2.20}$$

Suppose now that we actuate the kinematic chain with forces applied to the joints and that we want to find a similar decomposition. If we represent the linear relation between forces and torques in case of a constant lever arm with a function $\boldsymbol{g}(\cdot)$, we can write:

$$\begin{aligned} \mathcal{D}(\tilde{\boldsymbol{q}}(t)) &= a_1\boldsymbol{\tau}_1(t) + a_2\boldsymbol{\tau}_2(t) \\ &= a_1\boldsymbol{g}(\boldsymbol{f}_1(t)) + a_2\boldsymbol{g}(\boldsymbol{f}_2(t)) \\ &= \boldsymbol{g}(a_1\boldsymbol{f}_1(t) + a_2\boldsymbol{f}_2(t)) \end{aligned} \tag{2.21}$$

and we have found a linear decomposition in the forces $\boldsymbol{f}_1(t)$ and $\boldsymbol{f}_2(t)$ leading to the desired trajectory $\tilde{\boldsymbol{q}}(t)$. Suppose now that we actuate the kinematic chain with muscles producing a force in response to an activation input. The relation $\boldsymbol{h}(\cdot)$ between muscle activation $\boldsymbol{m}(t)$ and generated force $\boldsymbol{f}(t)$ is nonlinear and we have therefore:

$$\begin{aligned} \mathcal{D}(\tilde{\boldsymbol{q}}(t)) &= \boldsymbol{g}(a_1\boldsymbol{f}_1(t) + a_2\boldsymbol{f}_2(t)) \\ &= \boldsymbol{g}(a_1\boldsymbol{h}(\boldsymbol{m}_1(t)) + a_2\boldsymbol{h}(\boldsymbol{m}_2(t)) \\ &\neq \boldsymbol{g}(\boldsymbol{h}(a_1\boldsymbol{m}_1(t) + a_2\boldsymbol{m}_2(t))) \end{aligned} \tag{2.22}$$

In general it is not possible to find a decomposition in the activation inputs $\boldsymbol{m}_1(t)$ and $\boldsymbol{m}_2(t)$ by simply applying the superposition principle.

If we now think at $\boldsymbol{\tau}_i(t)$, $\boldsymbol{f}_i(t)$ and $\boldsymbol{m}_i(t)$ as primitives of a synergy-based controller it is clear that while the transition from torques to forces may not add any complexity in terms of linearity, the transition from forces to activations instead adds a new property which makes the whole system even more nonlinear and the job of the controller harder.

2.2.3 Torque model

In this model the kinematic chain is actuated by means of torques applied to the joints (Figure 2.2). The well-known dynamical equation describing this model is shown below and is taken from [67]:

$$\boldsymbol{D}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{E}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q}) = \boldsymbol{\tau} \tag{2.23}$$

where $\boldsymbol{q} = [q_e \quad q_s]^T$ is the configuration vector, with q_e and q_s the angular positions of the elbow and shoulder joints respectively and $\boldsymbol{\tau} = [\tau_e \quad \tau_s]^T$ the

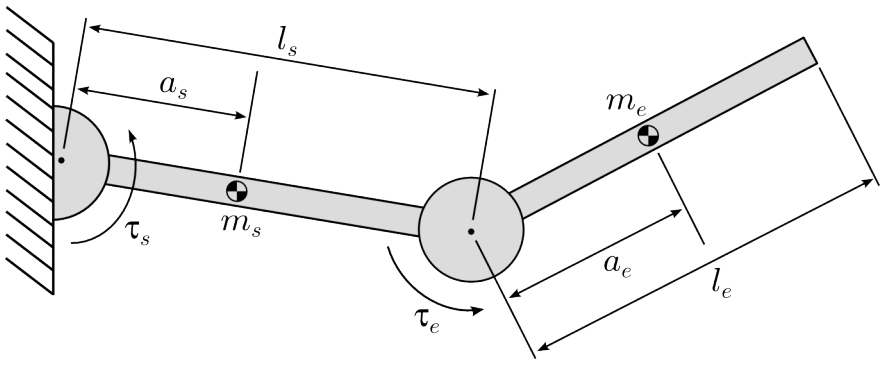


Figure 2.2: Torque actuated model. m_i are the segment masses, l_i their length, a_i the distance of the center of masses to the corresponding proximal joint, τ_i are the actuation torque.

torques applied to the respective joints. The matrices $\boldsymbol{D}$, $\boldsymbol{E}$ and $\boldsymbol{G}$ represent the inertial, centrifugal/coriolis and gravitational contributions respectively and are defined as follows:

$$\boldsymbol{D} = \begin{pmatrix} m_e a_e^2 + I_e & m_e a_e l_s \cos q_e + m_e a_e^2 + I_e \\ m_e a_e l_s \cos q_e + m_e a_e^2 + I_e & 2 m_e a_e l_s \cos q_e + m_e l_s^2 + m_e a_e^2 + I_e + m_s a_s^2 + I_s \end{pmatrix} \tag{2.24}$$

$$\boldsymbol{C} = \begin{pmatrix} 0 & m_e a_e l_s \sin q_e \dot{q}_s \\ -m_e a_e l_s \sin q_e (\dot{q}_e + \dot{q}_s) & -m_e a_e l_s \sin q_e \dot{q}_e \end{pmatrix} \tag{2.25}$$

$$\boldsymbol{G} = \begin{pmatrix} m_e g_0 a_e \cos(q_e + q_s) \\ m_e g_0 l_s \cos q_e + m_e g_0 a_e \cos(q_e + q_s) + m_s g_0 a_s \cos q_s \end{pmatrix} \tag{2.26}$$

where the quantities with the subscript e refer to the elbow joint and forearm segment, while those with subscript s refer to the shoulder joint and upper arm segment, as depicted in Figure 2.2. The quantities m_e and m_s are the masses of the two segments, l_e and l_s their length, a_e and a_s the positions of their centers of mass relative to the proximal joint, and I_e and I_s their moments of inertia relative to the centers of mass. The term g_0 is the gravitational acceleration, but because the kinematic chain is horizontal, gravity has no effect and the whole $\boldsymbol{G}$ matrix can be ignored.

2.2.4 Force model

The kinematic chain is actuated by means of 6 forces applied to the links, as depicted in Figure 2.3. In this configuration we have 4 forces acting on each

joint, 2 of them in place of single-joint muscles, and the other 2 in place of biarticular muscles.

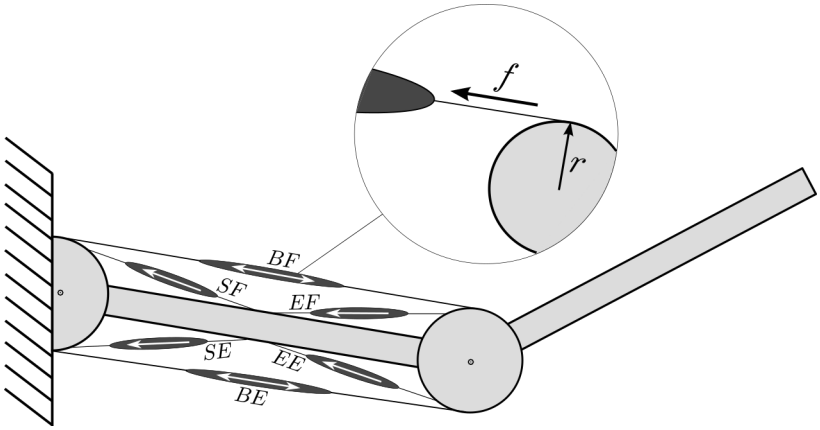


Figure 2.3: Forces actuated model. Each force f corresponds to a muscle in the musculoskeletal model and acts on a joint with lever arm r. The muscles are: Elbow Flexor (EF), Elbow Extensor (EE), Shoulder Flexor (SF), Shoulder Extensor (SE), Biarticular Flexor (BF), Biarticular Extensor (BE).

This type of configuration is widely used in literature (e.g. [46, 52, 45]). In particular Nijhof and Kouwenhoven [52] propose a procedure to lump 19 muscles of the human arm into 6 muscles organized as in Figure 2.3. Muscles origin and insertion points are taken from literature [72, 71, 4] and in part estimated from a skeleton model[2]. We used their data and lumping procedure, briefly illustrated in appendix A, to build our model. The lumping procedure results in the origins, insertion points and lever arms summarized in Table 2.2. Positive lever arms contribute to a positive torque and therefore to a flexion of the joint, while negative lever arms contribute to an extension. Coordinates are expressed in the reference frame of Figure 2.4, depicting the musculoskeletal model with the shoulder and elbow joints flexed 30°and 90°respectively. The x-axis runs along the Humerus while the y-axis is in parallel to the Ulna-Radius complex. The z coordinate of origin and insertion points was ignored because we were only interested in the horizontal plane[3].

We can express the actuation represented by the 6 forces with a vector

[2]The 19 muscles are those generating significant horizontal adduction/abduction of the shoulder and flexion/extension of the elbow [52]. The other arm muscles are not considered.

[3]By comparing the distance between insertion and origin in the three dimensional space with the distance calculated in the horizontal plane it can be verified that most muscles already lie essentially in the horizontal plane.

Table 2.2: Lumped muscles data obtained with the procedure described in [52]. $\tau_{s_{max}}$ and $\tau_{e_{max}}$ (Nm) are the maximal torques the muscle can generate and are obtained as sum of the single torques contributions from Table A.1. F_{max} (N) is the maximal force the muscle can generate, obtained dividing the maximal torque by the lever arm. For biarticular muscles the maximal forces is the mean value between the two maximal forces generated at the joints. O_x, O_y, I_x and I_y (cm) are the lumped origin and insertion point coordinates in the horizontal plane. r_s and r_e (cm) are the lever arms.

Muscle	$\tau_{\mathrm{s_{max}}}$	$\tau_{\mathrm{e_{max}}}$	F_{max}	$\mathrm{O_x}$	$\mathrm{O_y}$	$\mathrm{I_x}$	$\mathrm{I_y}$	$\mathrm{r_s}$	$\mathrm{r_e}$
SF	54.30	0	838	-5.6	10.9	8.7	1.6	6.5	0.0
SE	-43.08	0	1207	1.6	-5.6	9.4	0.4	-3.6	0.0
EF	0	37.63	1422	23.1	-0.6	30.9	14.4	0.0	2.6
EE	0	-28.23	1549	20.2	-1.7	32.4	2.0	0.0	-1.8
BF	7.21	14.08	304	-2.5	2.8	31.0	4.4	3.7	3.4
BE	-10.72	-12.06	469	-1.4	-2.8	32.6	-1.9	-3.2	-2.0

$\boldsymbol{F}$ defined as:

$$\boldsymbol{F} = \begin{pmatrix} f_{ef} & f_{ee} & f_{sf} & f_{se} & f_{bf} & f_{be} \end{pmatrix}^T \tag{2.27}$$

where the subscripts refer to the muscle names in Table 2.2 and depicted in Figure 2.3.

Each force acts on a joint through a lever arm. We consider a fixed lever arm, that is, it does not change with joint position. This assumption is not biologically plausible because we know that muscles slide over bones, joints and over each other and their lines of action changes constantly with the arm position. Nevertheless, this assumption let us investigate the implications of redundancy without adding further complexity.

If r_i is the lever arm associated to the force f_i, the total torque τ_j acting on a joint j is given by:

$$\tau_j = \sum_i r_i f_i \tag{2.28}$$

The total torques on elbow and shoulder joints of our model can therefore be written as follows:

$$\boldsymbol{\tau} = \begin{pmatrix} \tau_e \\ \tau_s \end{pmatrix} = \begin{pmatrix} r_{ef} f_{ef} + r_{ee} f_{ee} + r_{ebf} f_{bf} + r_{ebe} f_{be} \\ r_{sf} f_{sf} + r_{se} f_{se} + r_{sbf} f_{bf} + r_{sbe} f_{be} \end{pmatrix} = \boldsymbol{L} \begin{pmatrix} f_{ef} \\ f_{ee} \\ f_{sf} \\ f_{se} \\ f_{bf} \\ f_{be} \end{pmatrix} = \boldsymbol{LF} \tag{2.29}$$

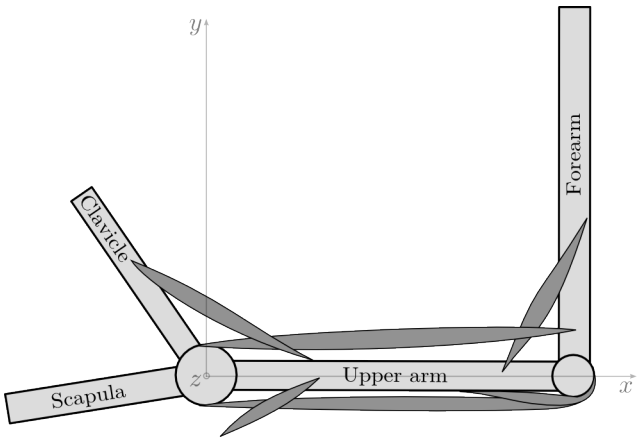


Figure 2.4: Skeletal model with the 6 muscles lumped following the procedure from [52].

The term $\boldsymbol{L} \in \mathbb{R}^{2\times 6}$ is the lever arm matrix with values given in Table 2.2:

$$\boldsymbol{L} = \begin{pmatrix} r_{ef} & r_{ee} & 0 & 0 & r_{ebf} & r_{ebe} \\ 0 & 0 & r_{sf} & r_{se} & r_{sbf} & r_{sbe} \end{pmatrix} \tag{2.30}$$

where each lever arm has the same subscript of the corresponding force, with the exceptions of r_{ebf} (r_{ebe}) and r_{sbf} (r_{sbe}) which are the lever arms of elbow biarticular flexor (extensor) and shoulder biarticular flexor (extensor).

The equations of motion (2.23) of the force model can therefore be rewritten as:

$$\boldsymbol{D}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{E}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q}) = \boldsymbol{\tau} = \boldsymbol{L}\boldsymbol{F} \tag{2.31}$$

The input to the kinematic chain are no more the torques $\boldsymbol{\tau}$ but the forces vector $\boldsymbol{F}$ containing the 6 independent inputs which actuate two joints. Thus, the system is redundant because the number of forces exceeds the number of joints. How is this redundancy exhibited? The lever arm matrix $\boldsymbol{L}$ plays a fundamental role. This matrix has rank 2 therefore its kernel it not empty[4]. Let's assume that the kinematic chain is actuated by a force $\boldsymbol{F}_1$ producing the torque $\boldsymbol{\tau}_1$ given by:

$$\boldsymbol{L}\boldsymbol{F}_1 = \boldsymbol{\tau}_1 \tag{2.32}$$

Since the $\ker(\boldsymbol{L}) \neq \{\emptyset\}$ we can find a force $\boldsymbol{F}_2 \neq \boldsymbol{0}$ such that $\boldsymbol{L}\boldsymbol{F}_2 = \boldsymbol{0}$. It follows that:

$$\boldsymbol{L}\boldsymbol{F}_1 = \boldsymbol{\tau}_1 = \boldsymbol{\tau}_1 + \boldsymbol{0} = \boldsymbol{\tau}_1 + \boldsymbol{L}\boldsymbol{F}_2 \Rightarrow \boldsymbol{L}(\boldsymbol{F}_1 - \boldsymbol{F}_2) = \boldsymbol{\tau}_1 \tag{2.33}$$

[4]By construction it has at least rank 2 because columns 1 and 3 or 2 and 4 are linearly independent, furthermore it has 2 rows. We conclude therefore that it must have rank 2.

hence we have found another force $\boldsymbol{F}_1 - \boldsymbol{F}_2$ leading to the same torque and to the same kinematic response. Given that the kernel is closed with respect to the multiplication by a scalar, there are infinite forces leading to the same actuation.

2.2.5 Activation model

The activation model is obtained from the force model by replacing the independent force inputs with muscles controlled in activation. Each muscle generates a force with the same line of action and lever arm as in the force model. The force configuration is therefore the same and all the considerations about redundancy are still valid. The independent inputs however are no more the forces, but the activations controlling the muscles.

In 1938 Hill showed that forces generated by muscle contractions depend on their internal state, represented by length and contraction velocity, and that the relation between activation and generated force is nonlinear. Thus, we can represent the activation model as in Figure 2.5, where $g(m, l, \dot{l})$ is the nonlinear function relating muscle activation m, state $(l, \dot{l})$, with l the muscle length, and generated force f.

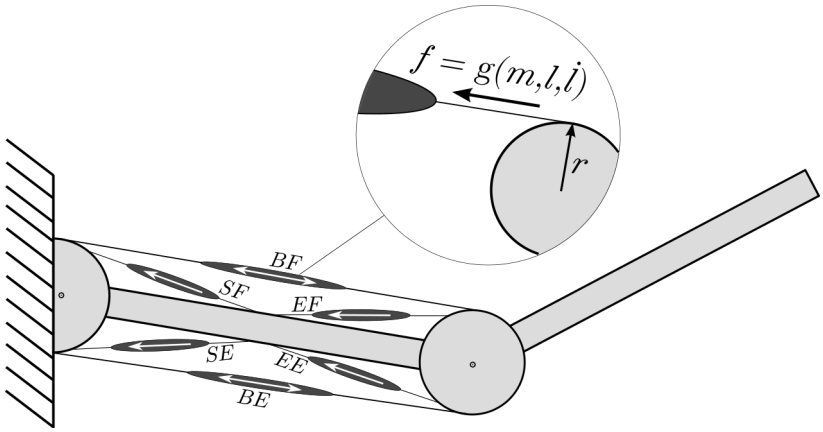


Figure 2.5: Activation actuated model. The force f produced by a muscle is a function of the its activation m, its length l and its contraction velocity $\dot{l}$. Each force acts on a joint with the lever arm r. The muscles are: Elbow Flexor (EF), Elbow Extensor (EE), Shoulder Flexor (SF), Shoulder Extensor (SE), Biarticular Flexor (BF), Biarticular Extensor (BE).

Muscle Dynamics

In the classic model from Hill, a muscle is composed by a *contractile element (CE)* in series with a *series element (SE)*, and a *parallel element (PE)*

arranged in parallel to both CE and SE[5]. The CE generates an active contraction force in response to an activation $m(t)$, which is a dimensionless quantity in the range $[0, 1]$. The generated force depends on muscle length and contraction velocity. The SE is a nonlinear spring-like element representing the passive elastic properties of muscle tissues and tendons. Being in series with the CE, it accounts for compliance during contractions; additionally its stiffness increases while stretching, resulting in increased joint stiffness during co-contractions [39, 70]. Finally the PE is a spring-like element which models the muscle viscoelasticity due to passive tissue inside and around the muscles (skin, fibers, vessels etc.).

The musculotendon series element (modeled by SE) is responsible for compliance in the muscle. Ignoring it corresponds to consider the muscle infinitely stiff and avoids to deal with differential terms in the muscle model. We decided therefore to focus on a model without the SE, which allows to analyze muscle nonlinearities without additional complexities. The model is depicted in Figure 2.6.

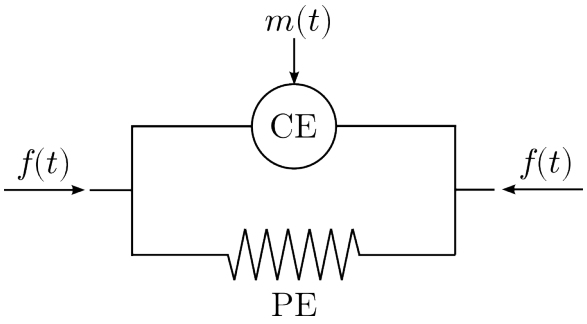


Figure 2.6: Muscle model composed by contractile element (CE) and a parallel element (PE) arranged in parallel. $m(t)$ is the muscle activation while $f(t)$ is the force resulting from both the CE and the PE.

In what follows we provide the details of the CE and the PE, and we show how they characterize muscle dynamics.

Contractile Element (CE) The force generated by the CE depends on muscle activation as well as on muscle length and velocity. In the following sections this force will be called *active force*. The relationship between active force, length and velocity is regulated by two equations, commonly called *force-length* and *force-velocity* relations. The force is generally considered

[5]Other models have been proposed in literature, for example with SE in series with both CE and PE. These models are described by other equations but are equivalent to the model described here.

instantaneous (not dependent on history) and is given by [69]:

$$f_{CE}(m, l, \dot{l}) = m f_l(l) f_v(\dot{l}) F_{max} \tag{2.34}$$

where $f_{CE}(m, l, \dot{l})$ is the force generated by the CE, m is the activation, $f_l(l)$ the force-length relation depending on the length l of the CE, $f_v(\dot{l})$ the force-velocity relation depending on the CE velocity $\dot{l}$, and F_{max} the maximal isometric force. The terms m, $f_l(l)$ and $f_v(\dot{l})$ accept values in the range $[0, 1]$, thus they act as weighting factors for the maximal isometric force.

Note that, since we have no SE in our model, the muscle length coincides with the length of CE and PE. We will therefore refer to this length simply as muscle length and we will use always the symbols l for the length ans $\dot{l}$ for the velocity.

Force-length relation This relation describes how muscle length affect the generated force, and originates from the fact that the muscle strength depends on the degree of overlapping of its fibers. According to [69] the force-length relation has its maximum at the muscle *optimum length* (about 1.05 times the rest length), and decreases for shorter or longer lengths. Most authors approximate this relation with a gaussian function ([52, 50, 39] and others). Accordingly, we chose to define the force-length relation as follows:

$$f_l(l) = e^{-\left(\frac{\frac{l}{L_{rest}} - L_{opt}}{k_l}\right)^2} \tag{2.35}$$

where l is the muscle length, L_{rest} the muscle rest length, L_{opt} the optimum length, and k_l the steepness of the gaussian curve (between 0 and 1).

Force-velocity relation Describes the dependency of the generated force on the muscle contraction velocity. For shortening muscles (contractions) the force increases for decreasing velocity, while it saturates at about 30% over the maximal isometric force for lengthening muscles (extensions). It is common practice to use two different formulations for contractions and extensions. For shortening muscles ($\dot{l} < 0$), the force-velocity relation is modeled with the following normalized hyperbolic formula (see [69]):

$$f_v(\dot{l})\Big|_{\dot{l}<0} = \frac{a_f(V_{max} - \dot{l})}{a_f V_{max} + \dot{l}} \tag{2.36}$$

where $\dot{l}$ is the contraction velocity, a_f the hyperbolic shape, and $V_{max} < 0$ the maximum unloaded contraction velocity. V_{max} and a_f depend on

muscle fiber composition [70]. According to [16], V_{max} should be scaled with activation, unless the movement being modeled requires low velocity of shortening muscles. Like the majority of the authors, we will not consider this scaling.

In case of lengthening muscles the force-velocity relation is such that the force saturates at about $F_s = 1.3F_{max}$ (see [69] and [52]):

$$f_v(\dot{l})\Big|_{\dot{l}>0} = \frac{F_s(a_f+1)\dot{l} + (F_s-1)a_f V_{max}}{(a_f+1)\dot{l} + (F_s-1)a_f V_{max}} \tag{2.37}$$

Note that both relations converge to 1 for $\dot{l} = 0$, that is, in isometric conditions.

Parallel Element (PE) The PE represents the passive elements of the muscle, and is commonly modeled as a nonlinear spring. It starts to have a non-negligible effect for muscle lengths higher than a certain threshold (typically about 30% over the rest length). Its effect is mostly linear in the primary range, but the steepness increases for higher values of muscle lengths, where the PE element starts to have big influence. Commonly, the force generated by the PE, called *passive force*, is modeled as an exponential function [36, 50, 52]:

$$f_{PE}(l) = F_{PE_{max}} \frac{e^{k_{PE}\frac{l}{L_{max}}} - 1}{e^{k_{PE}} - 1} \tag{2.38}$$

where $F_{PE_{max}}$ is the PE maximal force, k_{PE} is the steepness of the exponential and L_{max} is the maximum muscle length. In other words, the PE generates the force $F_{PE_{max}}$ when the muscle is at its maximum length. According to [64], depending on activation, the PE element can also push when compressed to extreme lengths and resist to the force produced by CE. We did not considered this possibility in our work.

Total muscle force Since the CE and the PE are arranged in parallel, the total force f is given by the sum of the two:

$$f(m,l,\dot{l}) = f_{CE}(m,l,\dot{l}) + f_{PE}(l) = m f_l(l) f_v(\dot{l}) F_{max} + f_{PE}(l) \tag{2.39}$$

Muscle Kinematics

In order to compute muscle lengths and velocities, a common assumption in literature (e.g. [59]) is that muscles roll over joints and slide over each

other. Hence, we can model the change in muscle length dl of a muscle as the product of the change in joint angles θ, times the lever arm r:

$$dl = rd\theta \tag{2.40}$$

To integrate this relation (i.e. to know the absolute lengths for a certain joint position) we need a reference position that associates given joint angles with the corresponding muscle lengths. The joint reference position $\hat{\boldsymbol{q}} = [\hat{q_e} \quad \hat{q_s}]^T$ that we have used corresponds to the arm posture described in section 2.2, where the elbow and shoulder joints are flexed at 90°and 30°respectively. The corresponding muscle lengths $\hat{\boldsymbol{l}} = [\hat{l}_{ef} \quad \hat{l}_{ee} \quad \hat{l}_{sf} \quad \hat{l}_{se} \quad \hat{l}_{bf} \quad \hat{l}_{be}]^T$ were computed as the euclidean distance between lumped origin and insertion coordinates (these coordinates are reported in Table 2.2).

For a single muscle, the absolute length l is given by:

$$l = \hat{l} + r_e(\hat{q}_e - q_e) + r_s(\hat{q}_s - q_s) = \hat{l} + \begin{pmatrix} r_e & r_s \end{pmatrix} \begin{pmatrix} \hat{q}_e \\ \hat{q}_s \end{pmatrix} - \begin{pmatrix} r_e & r_s \end{pmatrix} \begin{pmatrix} q_e \\ q_s \end{pmatrix} \tag{2.41}$$

where r_e and r_s are the lever arms at the elbow and shoulder joint respectively and $\boldsymbol{q} = (q_e \quad q_s)^T$ is the configuration state. If the muscle is not biarticular, then either r_e or r_s are 0. Eq. (2.41) can be rewritten in matrix form as follows

$$\boldsymbol{l} = \hat{\boldsymbol{l}} + \boldsymbol{L}^T(\hat{\boldsymbol{q}} - \boldsymbol{q}) \tag{2.42}$$

where $\boldsymbol{l} = [l_{ef} \quad l_{ee} \quad l_{sf} \quad l_{se} \quad l_{bf} \quad l_{be}]^T$ is the vector of muscle lengths and $\boldsymbol{L}$ is the lever arm matrix defined in Eq. (2.30). The expression to calculate muscles velocities can be calculated similarly, or simply by deriving Eq. (2.42):

$$\dot{\boldsymbol{l}} = -\boldsymbol{L}^T\dot{\boldsymbol{q}} \tag{2.43}$$

Muscle Parameters

In this section we define the values of the muscle parameters described in the previous sections.

Is is difficult to estimate plausible values for those parameters because each muscle has different properties and some of them also change from individual to individual. Literature does not provide ready to use data and, additionally, we work on a lumped musculoskeletal model. We believe, however, that for our purpose it is not important to obtain a quantitatively accurate model of the muscle. Rather, we want to make sure that the model resembles qualitatively properties and dynamics of real muscles. We chose therefore a common parameter set for all muscles.

Rest length (L_{rest}) We use the musculoskeletal arm configuration described in 2.2, with the elbow and shoulder flexed 90°and 30°, as resting position. Starting from the resting position, we compute muscle rest lengths as the euclidean distances between origin and insertion points in Table 2.2.

Maximum length (L_{max}) Is calculated by placing the musculoskeletal model in the 4 extreme joint positions. Relation (2.42) is then used to calculate the muscle lengths at these positions, and the maximum value for each muscle is taken as maximum length.

Optimum length (L_{opt}) In accordance with [69] we take 1.05. This means that muscles can express their maximal force when their length is $1.05L_{rest}$.

Force-length relation gaussian steepness (k_l) According to [69], the force-length relation has its maximum at the muscle optimum length L_{opt}, and tends to 0 at about 0.4 and 1.5 times the muscle rest length. Such a profile can be obtained with $k_l = 0.50$.

Force-velocity relation hyperbolic shape (a_f) Depending on the muscle, this parameter takes a value in the range $[0, 1]$, and is generally below 0.25 for slow fiber muscles and higher for faster muscles [69]. According to [70], for example, elbow extensors and flexors have values between 0.30 and 0.45. We chose $a_f = 0.45$ for all muscles.

Maximal unloaded contraction velocity (V_{max}) Different authors propose a variety of values for this parameter. For Winters [69] it depends on the fiber types composing the muscles and ranges from $2L_0/sec$ to $8L_0/sec$ where L_0 is the mean fiber rest length. In [68] the author proposes values above $10L_0/sec$. According to [37], L_0 takes values between $0.33L_{rest}$ for distal muscles (i.e. elbow muscles) and $0.75L_{rest}$ for other muscles. We chose $L_0 = 0.54L_{rest}$ for all muscles and $V_{max} = 5L_0/sec = 2.7L_{rest}/sec$. These values lay in the middle of the ranges suggested in literature.

Maximal isometric force (F_{max}) This parameter is taken from our lumped musculoskeletal model described in section 2.2. Values are shown in Table 2.2.

Maximal PE force ($F_{PE_{max}}$) According to [64], the maximum PE force is less than 10% of F_{max} and is obtained at maximum anatomical length. Maximum anatomical length is the maximum length a muscle can have across all anatomical configurations of the joints it crosses [64]. This value can be

estimated by subtracting the tendon length from the entire musculotendon length. Because we are not considering tendons (no SE element), we take the maximum anatomical length as L_{max} and set $F_{PE_{max}} = 0.1F_{max}$.

PE force steepness (k_{PE}) A value which fits the curve description given in literature is $k_{PE} = 20$.

Figure 2.7 shows the normalized profiles of the CE force-velocity and force-length relations, and of the PE force function for the described parameters. Figure 2.8 shows the overall normalized muscle force as a function of its length and velocity. In both figures, length is normalized with respect to L_{rest}, velocity with respect to V_{max}, and PE force with respect to $F_{PE_{max}}$.

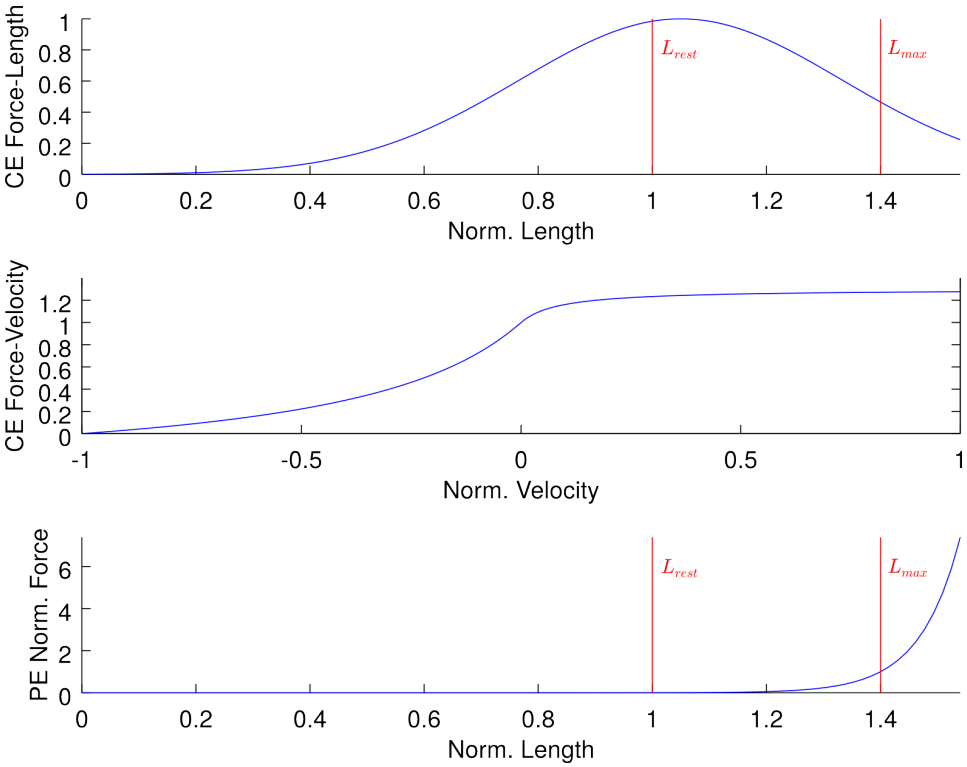


Figure 2.7: CE and PE normalized characteristic profiles. Length is normalized with respect to L_{rest}, velocity with respect to V_{max} and PE force with respect to $F_{PE_{max}}$.

Kinematic chain activation dynamics

Now that we have defined the dynamics of a single muscle as interaction between the CE and PE elements, we derive the dynamical equations for the whole kinematic chain. As we have done for the force model, the activation

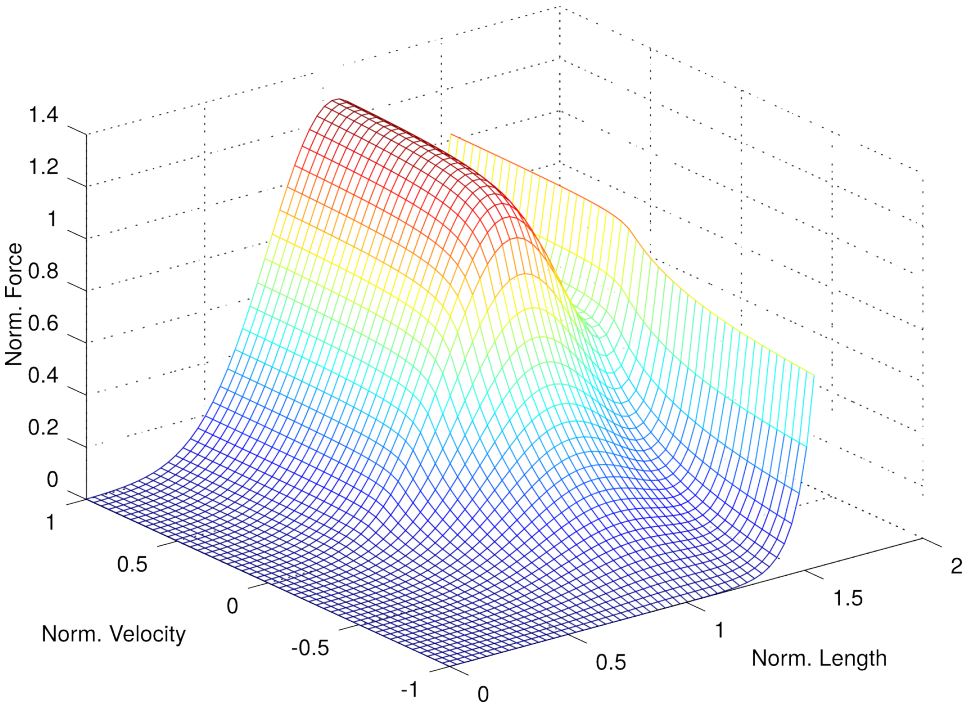


Figure 2.8: Three-dimensional surface representing muscle force as a function of length and velocity. Length is normalized with respect to L_{rest}, velocity with respect to V_{max}, and the resulting force with respect to the maximal isometric force F_{max}.

input can be arranged in a vector $\boldsymbol{M}$:

$$\boldsymbol{M} = \begin{pmatrix} m_{ef} & m_{ee} & m_{sf} & m_{se} & m_{bf} & m_{be} \end{pmatrix}^T \tag{2.44}$$

By using Eq. (2.39) we can define the vector $\boldsymbol{F}$ of the forces generated by the muscles as:

$$\boldsymbol{F} = \boldsymbol{C}(\boldsymbol{l}, \dot{\boldsymbol{l}})\boldsymbol{M} + \boldsymbol{P}(\boldsymbol{l}) \tag{2.45}$$

where $\boldsymbol{l} = [l_{ef} \dots l_{be}]^T$ and $\dot{\boldsymbol{l}} = [\dot{l}_{ef} \dots \dot{l}_{be}]^T$ are the muscle length and velocity vectors. The term $\boldsymbol{C}(\boldsymbol{l}, \dot{\boldsymbol{l}})$ is a diagonal matrix that represents the CE: it contains at position i of its diagonal the maximal isometric force weighted by the force-length and force-velocity relations of muscle i:

$$\boldsymbol{C}(\boldsymbol{l}, \dot{\boldsymbol{l}}) = \begin{pmatrix} f_{l_{ef}}(l_{ef}) f_{v_{ef}}(\dot{l}_{ef}) F_{max_{ef}} & & \\ & \ddots & \\ & & f_{l_{be}}(l_{be}) f_{v_{be}}(\dot{l}_{be}) F_{max_{be}} \end{pmatrix} \tag{2.46}$$

The vector $\boldsymbol{P}(\boldsymbol{l})$ is the contribution of the PE:

$$\boldsymbol{P}(\boldsymbol{l}) = \begin{pmatrix} f_{PE_{ef}}(l_{ef}) \\ \vdots \\ f_{PE_{be}}(l_{be}) \end{pmatrix} \tag{2.47}$$

The expression of the total torque applied to the joints follows from relations (2.29) and (2.45):

$$\boldsymbol{\tau} = \boldsymbol{L}\boldsymbol{F} = \boldsymbol{L}(\boldsymbol{C}(\boldsymbol{l}, \dot{\boldsymbol{l}})\boldsymbol{M} + \boldsymbol{P}(\boldsymbol{l})) \tag{2.48}$$

where $\boldsymbol{L}$ is the lever arm matrix. The equations of motion for the activation model can be rewritten as:

$$\boldsymbol{D}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{E}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q}) = \boldsymbol{L}\boldsymbol{C}(\boldsymbol{l}, \dot{\boldsymbol{l}})\boldsymbol{M} + \boldsymbol{L}\boldsymbol{P}(\boldsymbol{l}) \tag{2.49}$$

2.3 Experimental setting

In this section we describe the procedures followed to execute our experiments, and the methods used in order to evaluate the obtained results. We actuated our kinematic chain with linear combinations of primitives representing synergies, as described in section 2.1.3, and have measured their performance in solving reaching tasks, as described in section 2.1.4. These experiments have been performed in all the three models described in sections 2.2.3, 2.2.4 and 2.2.5.

DRD Framework We have conducted our experiments in the DRD framework described in section 2.1. In our setting the differential operator $\mathcal{D}$ represents our kinematic chain, the vector $\boldsymbol{q}(t) = [q_e(t) \quad q_e(t)]^T$ its configuration and $\boldsymbol{u}(t)$ the actuation, which, depending on the model, is a torque, a force or an activation.

The synergies $\boldsymbol{\Phi} = [\boldsymbol{\phi}_1(t) \dots \boldsymbol{\phi}_{n_\phi}(t)]$ are defined in the actuation space of the kinematic chain (torques, forces or activations). The dynamic responses $\boldsymbol{\Theta} = [\boldsymbol{\theta}_1(t) \dots \boldsymbol{\theta}_{n_\theta}(t)]$ where $\boldsymbol{\theta}_i(t) = [\theta_{e_i} \quad \theta_{s_i}]^T$ are the responses of the kinematic chain to the synergies. In this setting $n_\phi = n_\theta = N$. The vector $\boldsymbol{b} = [b_1 \dots b_N]^T$ contains the synergy mixing coefficients. For the rest of this manuscript we will identify all these terms with different symbols depending on the model. These symbols are listed in Table 2.3.

Task definition We defined reaching tasks as movements from an initial joint position $\boldsymbol{q}_0 = [q_{e_0} \quad q_{s_0}]^T$ at time $t = 0$ to a final position $\boldsymbol{q}_f =$

Table 2.3: Symbols used in the torque, force and activation model.

Model	Actuation	Synergy	All Synergies	Mixing coeff.
Torque Model	$\boldsymbol{\tau}$	$\boldsymbol{\phi}$	$\boldsymbol{\Phi}$	$\boldsymbol{a}$
Force Model	$\boldsymbol{F}$	$\boldsymbol{\gamma}$	$\boldsymbol{\Gamma}$	$\boldsymbol{b}$
Activation Model	$\boldsymbol{M}$	$\boldsymbol{\psi}$	$\boldsymbol{\Psi}$	$\boldsymbol{c}$

$[q_{e_f} \quad q_{s_f}]^T$ at time $t = T$. Both initial and final velocities are null and acceleration is unconstrained. While $\boldsymbol{q}_0$ is fixed across tasks, $\boldsymbol{q}_f$ defines each task instance. The task is therefore defined by the following constraints:

$$\begin{aligned} \boldsymbol{q}(0) &= \boldsymbol{q}_0, \quad \dot{\boldsymbol{q}}(0) = \boldsymbol{0} \\ \boldsymbol{q}(T) &= \boldsymbol{q}_f, \quad \dot{\boldsymbol{q}}(T) = \boldsymbol{0} \end{aligned} \tag{2.50}$$

The alternant and task constraint matrices are defined as follows (see Eq. (2.8)):

$$\boldsymbol{M} = \begin{pmatrix} \boldsymbol{\theta}_1(0) & ... & \boldsymbol{\theta}_{n_\theta}(0) \\ \boldsymbol{\theta}_1(T) & ... & \boldsymbol{\theta}_{n_\theta}(T) \\ \dot{\boldsymbol{\theta}}_1(0) & ... & \dot{\boldsymbol{\theta}}_{n_\theta}(0) \\ \dot{\boldsymbol{\theta}}_1(T) & ... & \dot{\boldsymbol{\theta}}_{n_\theta}(T) \end{pmatrix} \quad \boldsymbol{P} = \begin{pmatrix} \boldsymbol{q}_0 \\ \boldsymbol{q}_f \\ \boldsymbol{0} \\ \boldsymbol{0} \end{pmatrix} \tag{2.51}$$

Each element of the two matrices is a two dimensional vector with the elbow and shoulder joint coordinates. The system of equations (2.8) is therefore composed in this case by 8 linear equations and a number of unknowns depending on the number of synergies/dynamics responses.

The time interval $[0 \quad T]$ of the reaching tasks is fixed with $T = 750$ms. This value was chosen as the mean duration of reaching movements in humans as reported from literature [40, 5, 46].

Synthesis of synergies Alessandro et al. [2] propose a method to synthesize synergies using the DRD framework. They hypothesize that the synergies involved in the solution of a certain class of tasks (e.g. reaching tasks) are themselves solutions to tasks of the same type. A *class of task* is an equivalence relation between tasks which share common properties. For example, while the class of reaching tasks may be composed by all reaching tasks with initial and final velocity zero, the class of via-point tasks may be instead composed by all tasks which trajectory passes through a certain via-point, expressed as task constraint. The idea behind this hypothesis is that synergies picked in a class of tasks embed information about that class and are more suitable to solve other task instances of similar type.

The method is composed by two steps, the *exploration* followed by the *reduction*. In the exploration phase the dynamical system (i.e. the kinematic chain) is actuated with a extensive set of actuations $\mathbf{\Phi}^*$, starting from different initial conditions. This results in a corresponding set of DRs $\mathbf{\Theta}^*$ uniformly distributed in the phase space. Actuations and initial conditions are not restricted to a particular type. In the reduction phase the trajectories of a small set of tasks, called *prototasks*, are found by interpolating on the exploration DRs $\mathbf{\Theta}^*$, as in Eq. (2.6). The actuations solving the prototasks are found by inverse dynamics from the trajectories (see Eq. (2.9)), and are taken as the synergy set $\mathbf{\Phi}$. The number of prototasks will therefore correspond to the number of synergies.

The prototasks are defined by task constraints and are added iteratively with the following procedure:

1. define the first prototask. No particular restrictions are applied. In the case of reaching tasks for example, the first prototask may be defined by a final position $\boldsymbol{q}_f$ somewhere in the configuration space of the kinematic chain, by the initial configuration $\boldsymbol{q}_0$ and by null initial and final joint velocities;

2. solve the newly added prototask by interpolation on $\mathbf{\Theta}^*$, and take its corresponding actuation, found by inverse dynamics, as synergy, adding it to the initially empty set $\mathbf{\Phi}$;

3. define a sufficient number of test tasks (e.g. reaching tasks) uniformly distributed and solve them by linearly combining the synergies $\mathbf{\Phi}$ as described in section 2.1;

4. for each task evaluate the projection error. This leads to a mapping between the tasks and the synergies performance in approximating their solutions;

5. if the overall performance is not satisfactory add a new prototask where the error is maximal and reiterate from step 2, otherwise terminate. In other words, the newly added prototask has similar constraints to the tasks for which the solutions are poorly approximated by the synergies.

We used the method described above in order to synthesize a set of synergies for our models and used them to evaluate the performance and the effect of redundancy and nonlinearity on the control of reaching tasks by linear combination of synergies.

In the exploration phase we actuated the torque model with a set of about 120 random torque profiles leading to 120 exploration dynamic responses.

These actuations were generated by filtering (with a 9th-order low pass butterworth filter) random numbers in the interval of feasible torques ([-4.4, +4.4] Nm for shoulder, and [-1.3, +1.3] Nm for elbow [31][6]). For each torque we chose a random initial condition (initial joint configuration) and we simulated the system; if the obtained joint trajectory was outside the joint boundaries, both DR and actuation were discarded. The result of the exploration is shown in Figure 2.9.

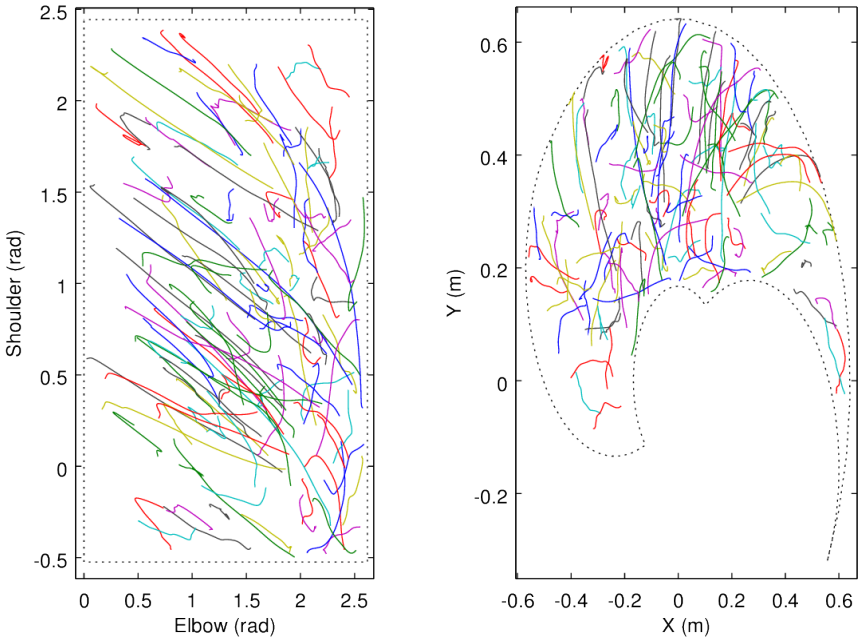


Figure 2.9: Exploration dynamic responses represented in joint space (left) and end effector space (right). Each colored line is a different dynamic response.

In the reduction phase we used the exploration DRs in order to find the synergies as solutions to prototasks. Following the hypothesis of Alessandro et al. [2] (synergies involved in the solution of a task are themselves solutions to tasks in the same class) we picked prototasks from the class of reaching tasks with constrained initial and final positions and velocities. We iterated the reduction procedure at least 12 times, to find at least 12 synergies. At each step we evaluated the performance of the synergies found till then.

Performance assessment The performance of the synthesized synergies was evaluated on a wide set of reaching tasks. We defined a mesh of more

[6]Values taken from Fig 3A and 3B in [31].

than 800 points uniformly distributed in the whole configuration space and have taken each point as final configuration $\boldsymbol{q}_f$ of one task instance. The initial configuration $\boldsymbol{q}_0$ was fixed. We solved each task with the set of the first n synergies found during the reduction, for $n = 1 \dots 12$. For each n we calculated the mean (relative) projection error and the mean forward dynamics error over all the tasks, and plotted these error against the number of synergies n. This resulted in two plots, the *(relative) projection error vs. number of synergies* and the *forward dynamics error vs. number of synergies*. These plots give information on the ability of the synergies in approximating task actuations, and on the quality of the trajectory obtained from the approximated solutions. They also show how the overall performance scales by adding new synergies, and make possible a good and intuitive comparison between the torque, the force and the activation models.

Chapter 3

Results: Torque Model

In this chapter we present the results obtained on the torque actuated kinematic chain. We show the prototasks found using the reduction procedure described in 2.3 and show that a small number of synergies is capable of controlling the kinematic chain with good performance, leading to a considerable dimensionality reduction of the controller. The same experiments on a slightly different 2-joints kinematic chain actuated with torques have already been done by Alessandro et al. in [2]. We compare their results with those obtained in our work and use the latter as performance reference mark for the force and activation models.

3.1 Reduction

The result of the reduction procedure on the torque model, iterated 12 times in order to synthesize from 1 to 12 synergies, is shown in Figure 3.1. Each panel corresponds to one iteration (ordered in row-major order), and shows the projection error (color coded) obtained by solving more than 800 reaching tasks distributed in the operational space of the arm. The darker the color, the smaller the error. In iteration 5 (fifth image in row-major order, starting from the top-left image), for example, the color corresponds to an error of 10^{-1} in most areas, with some darkening around the prototasks (10^{-2}) and a slightly brighter area in the center (between 1 and 10^{-1}). The new prototask (no. 6) is placed in this area, because there the projection error is higher. The addition of prototask no. 6 results in a darkening in the next panel, meaning that the projection error is smaller and performance is increased. The 800 test tasks are then evaluated again and the resulting performance is used to add prototask no. 7. The procedure reiterates till prototask no. 12 is added and all test tasks are evaluated again for the last

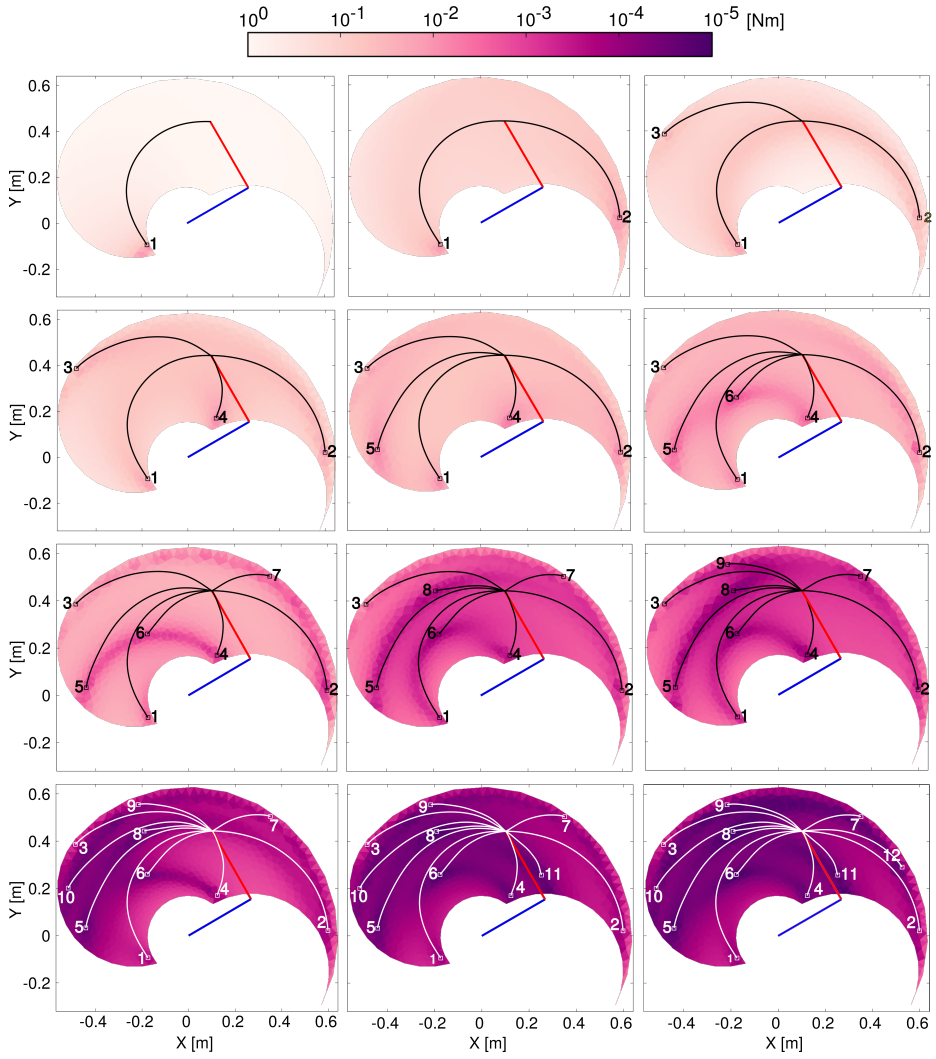


Figure 3.1: Results of the reduction procedure for the torque actuated kinematic chain. Each panel corresponds to a number of synergies. Each point of the operational space is colored depending on the performance of the synergies in approximating a reaching task solution with final position in that point. Bright areas correspond to high projection errors while dark areas correspond to small projection errors as depicted in the color bar above.

time (bottom-right image).

Table 3.1 shows the prototask coordinates expressed in elbow and shoulder joint angular positions.

Table 3.1: Prototasks positions in the torque model, expressed as elbow and shoulder joint angles.

#	q_e [rad]	q_s [rad]
1	2.5183	2.2611
2	0.7411	-0.3497
3	0.5114	2.2047
4	2.4789	-0.4042
5	1.6219	2.2180
6	2.1195	1.0429
7	0.5884	0.6575
8	1.4434	1.2242
9	0.7642	1.5458
10	1.0547	2.2218
11	1.9514	-0.2364
12	0.7256	0.1289

3.2 Performance

Figure 3.1 gives an idea of the performance obtained when solving reaching tasks by linear combination of synergies. Such results can be summarized as in Figure 3.2, where each point represents the mean projection error across the test reaching tasks for each number of synergies (i.e. for each iteration of the reduction). As it can be seen, performance increases by adding new synergies. For example, while 4 synergies approximate a desired actuation with a mean error below $10^{-1}Nm$, with 8 synergies the mean error approaches $10^{-3}Nm$. Figure 3.3 shows the actuations and kinematic trajectories obtained by approximating the desired control signal with 8 synergies for the task with the highest projection error. The desired actuation obtained from the inverse dynamics of the desired trajectory, and the trajectory itself, are very close to those obtained from the linear combination of synergies.

What does all this means in terms of end effector movement? Will the kinematic chain controlled by linear combination of synergies be able to reach the desired configuration? The plot of the end effector forward dynamics error vs. number of synergies (see section 2.3) in Figure 3.4 shows that the forward dynamics performance scales similarly to the ability of the synergies to approximate the desired actuation (measured by the projection error). With 8 synergies the mean forward dynamics error is less than 10^{-3},

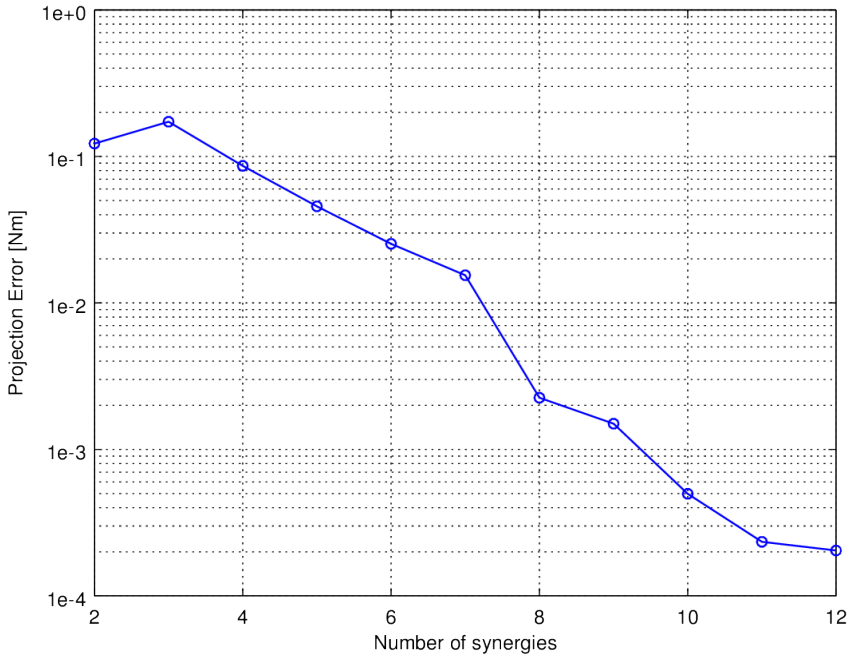


Figure 3.2: Projection error vs. number of synergies for the torque model.

meaning that the kinematic chain can reach the desired target with an error smaller than $1mm$[1].

As observed in [2], a traditional controller would have to "choose" joint torques at each timestamp. In contrast the 8 synthesized synergies allow the controller to generate satisfactory actuations by choosing only 8 scalars, i.e. the mixing coefficients (see Eq. (2.2)). Hence, 8 synergies represent a very high dimensionality reduction.

3.3 Comparison with previous work

The obtained results are similar to that reported in [2] for point-to-point tasks with null initial and final velocity. In their work, the authors obtained with 7 synergies a mean projection error $< 10^{-2}Nm$ and a mean forward dynamics error of ca. 10^{-2}. Here we achieve a little bit higher but still similar projection error with the same number of synergies, and, by adding new

[1]The forward dynamics error is calculated from both position and velocity trajectory components, measured in m and m/s respectively (see definition in section 2.1). As a result, if the forward dynamics error is 10^{-3}, the error in position cannot be higher than $1mm$.

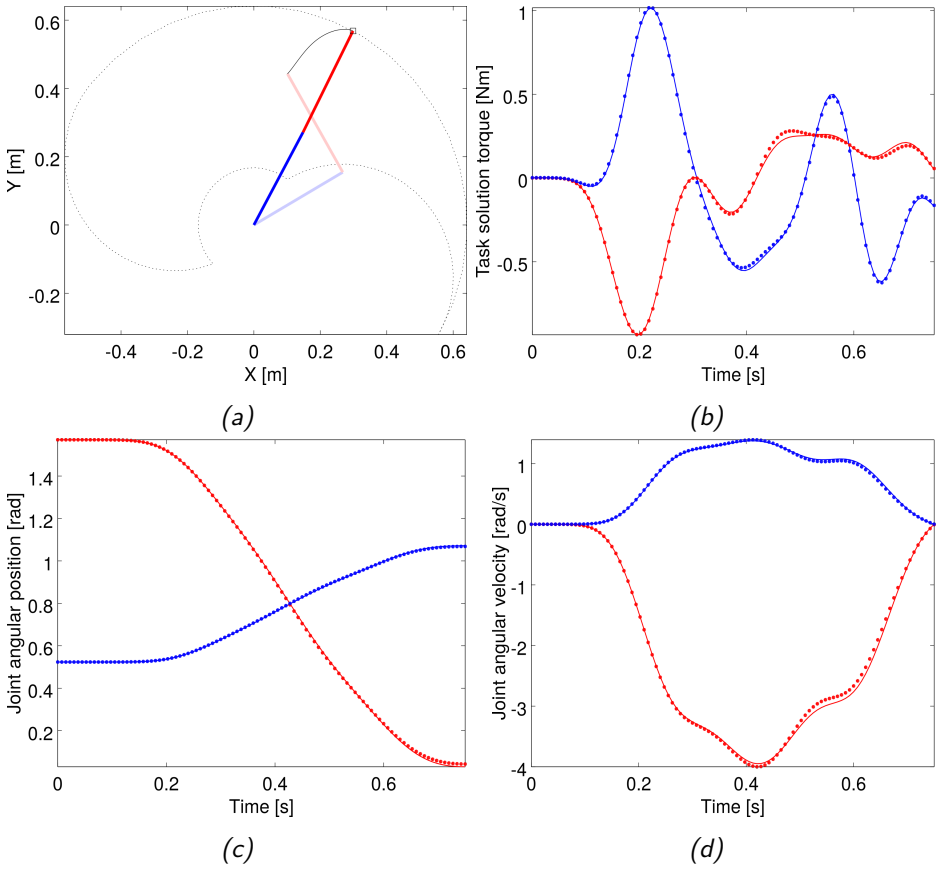


Figure 3.3: End effector trajectory (a), torques actuation (b), joint positions (c) and joint velocities (d) obtained by linearly combining the first 8 synergies found during the reduction for the task with the highest projection error. Red lines refer to the elbow joint while blue lines to the shoulder joint. Continuous lines in (b) represent the desired actuations (computed by evaluating the inverse dynamics on the desired trajectory) while dotted lines are the closest actuations in the linear span of the synergies. Continuous lines in (c) and (d) represent the desired trajectories (linear combinations of DR, see Eq. (2.6)). Dotted lines represent the trajectories obtained when actuating the kinematic chain by means of linear combination of synergies.

synergies, performance scale very similarly. This can be verified comparing Figure 3.2 with the similar plot reported in [2]. The small differences in the projection error can be due to a variety of factors (e.g. choice of prototasks, different kinematic chain properties etc.) and are not relevant. As shown in Figure 3.4 instead, the forward dynamics error with 7 synergies is ca. 10^{-2}, perfectly in line with that reported in [2].

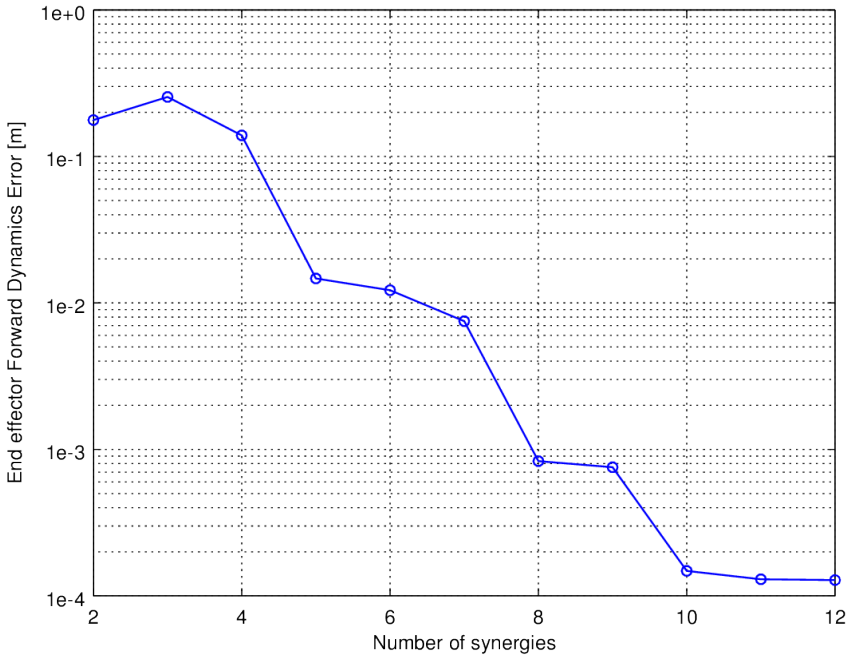


Figure 3.4: End effector forward dynamics error vs. number of synergies for the torque model.

3.4 Torque model as performance reference

The performance obtained on the torque model will be used as a reference to compare the results obtained on the force model (which adds redundancy) and on the activation model (which adds both redundancy and muscle non-linearities). These results will be presented in the next chapters.

Chapter 4

Results: Force Model

In this chapter we present the results obtained when controlling the kinematic chain by means of redundant forces applied to the links (they translate into torques on the joints). In addition to redundancy, we also introduce biological constraints (i.e forces ≥ 0). We show with examples that the problem of solving the inverse dynamics of the model is non-trivial. We then formalize this problem mathematically and demonstrate that by choosing a certain criteria it is possible to obtain valid force inputs, and that the number of synergies needed to obtain results comparable to that of the torque model does not increase, despite redundancy and constraints.

4.1 Torque to force relation

In order to compute both synergies and task solutions we use the inverse dynamics of the model to be controlled. In the force model this means that given a certain trajectory in kinematic space we need a strategy to find a force profile which leads to that trajectory. In the torque model the correspondence between kinematic trajectory and torque is one-to-one, that is, there is only one torque profile leading to a given trajectory, and it can be calculated by means of the closed form (2.23). In the force model instead, because of the redundancy, different forces lead to the same torque and therefore to the same trajectory, as illustrated in Figure 4.1. The objective consists therefore in finding a *torque-to-force* relation $\boldsymbol{g}(\cdot)$ from torque space $\mathrm{T} \subseteq \mathbb{R}^2$ to force space $\mathcal{F} \subseteq \mathbb{R}^6$ such that:

$$\boldsymbol{g} : \mathrm{T} \to \mathcal{F} \quad : \boldsymbol{\tau}(t) = \boldsymbol{L}\boldsymbol{g}(\boldsymbol{\tau}(t)) = \boldsymbol{L}\boldsymbol{F}(t) \quad \forall \boldsymbol{\tau}(t) \in \mathrm{T} \tag{4.1}$$

where $\boldsymbol{\tau}(t)$, $\boldsymbol{F}(t)$ and $\boldsymbol{L}$ are torque, force and the lever arm respectively, already introduced in section 2.2.4. Since under our assumptions synergies

are solutions to prototasks, there will also be different synergies leading to the same dynamic response.

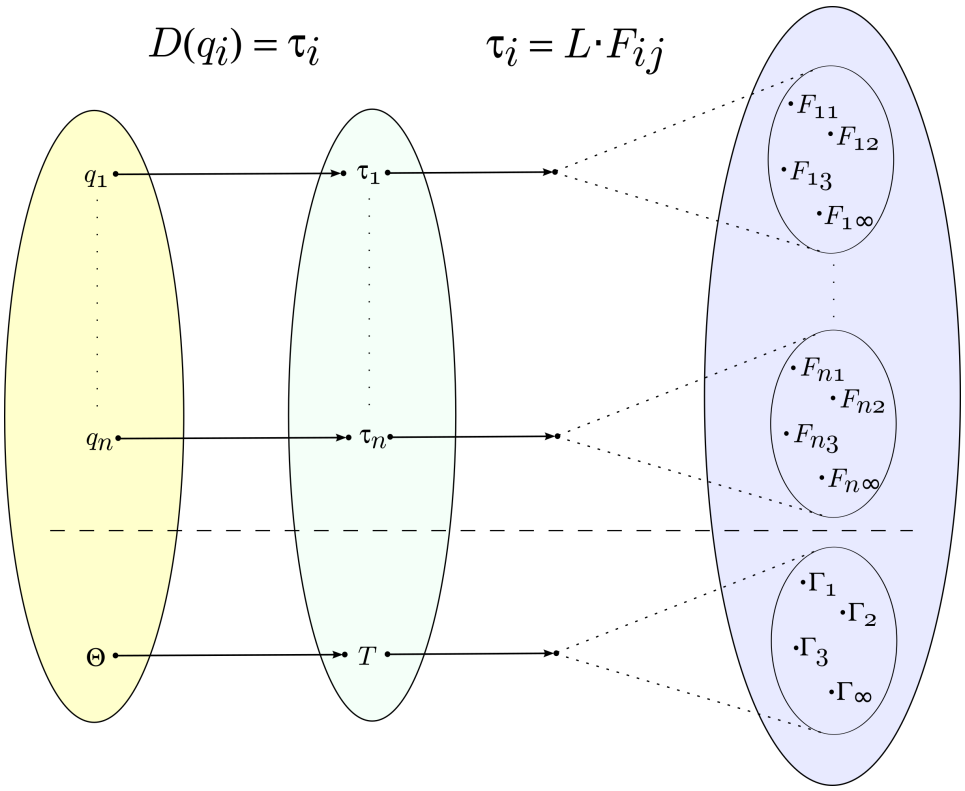


Figure 4.1: Redundancy in the force model. While the correspondence between joint kinematic trajectory $\boldsymbol{q}_i$ and torque $\boldsymbol{\tau}_i$ is one-to-one, the correspondence between torque and force $\boldsymbol{F}_{ij}$ is one-to-many. Since synergies are solutions to prototasks the correspondence between torque actuations $\mathbf{T}$ and force synergies $\mathbf{\Gamma}$ is one-to-many as well.

In addition to redundancy, the choice of the torque-to-force relation is also influenced by biological constraints. In our setting forces represent muscle outputs. Thus they must be physiologically plausible. In the following we briefly discuss the two fundamental properties characterizing the force model and the torque-to-force relation and introduce the math necessary to formalize them.

4.1.1 Redundancy

In section 2.2.4 we showed that redundancy is exhibited through the lever arm matrix $\boldsymbol{L}$ and we demonstrated that an infinite number of forces can lead to a given torque. We will formalize the mathematical relation between forces and torques in the next section. For now we show, with the example

in Figure 4.2, that two completely different forces lead to the same torque. Each force can be decomposed in two components: one that contributes to the torque and the other one that doesn't. If two different force profiles share the first of these components, they will lead to the same torque, independently on the second component. Since synergies are basis which are linearly combined together in order to approximate a set of input forces, their profile is extremely important and influences the model performance. In approximating task solutions the non-contributing component of the force synergies can be changed in order to obtain good performance, without affecting the resulting actuation.

4.1.2 Biological constraints

In the force model forces are assumed to be generated by muscles. Thus, they must be physiologically plausible. In particular muscles can only pull, meaning that they contribute to the torque in only one direction, either positive (flexion) or negative (extension). Since the direction is already encoded in the sign of the corresponding element(s) of the lever arm matrix $\boldsymbol{L}$, $\boldsymbol{F}(t)$ must be constrained to be positive. Additionally, muscle forces are bounded. The maximal force $\boldsymbol{F}_{max}$ is a property of our lumped musculoskeletal model and is reported in Table 2.2. The torque-to-force relation in 4.1 can therefore be rewritten as:

$$\boldsymbol{g} : \mathrm{T} \to \mathcal{F} \quad : \boldsymbol{\tau}(t) = \boldsymbol{L}\boldsymbol{g}(\boldsymbol{\tau}(t)) = \boldsymbol{L}\boldsymbol{F}(t) \wedge \ \boldsymbol{g}(\boldsymbol{\tau}(t)) \in [\boldsymbol{0}, \boldsymbol{F}_{max}] \quad \forall \boldsymbol{\tau}(t) \in \mathrm{T} \tag{4.2}$$

4.2 Nullspace

In this section we formalize the torque-to-force relation introduced in the previous paragraphs.

Each force $\boldsymbol{F}_i(t) \in \mathbb{R}^6$ can be decomposed in two components, one contributes to the torque and the other one doesn't. The minimum-norm component $\boldsymbol{F}_i^0(t) \in \mathbb{R}^6$ leading to a specific torque $\boldsymbol{\tau}_i(t) \in \mathbb{R}^2$ can be computed as:

$$\boldsymbol{F}_i^0(t) = \boldsymbol{L}^+ \boldsymbol{\tau}_i(t) \tag{4.3}$$

where $\boldsymbol{L}^+ \in \mathbb{R}^{6\times 2}$ is the Moore-Penrose pseudoinverse of $\boldsymbol{L}$. The non-contributing component $\boldsymbol{F}_i^k(t) \in \mathbb{R}^6$ is in the nullspace of $\boldsymbol{L}$ and can be written as:

$$\boldsymbol{F}_i^k(t) = \boldsymbol{N}\boldsymbol{w}_i(t) \tag{4.4}$$

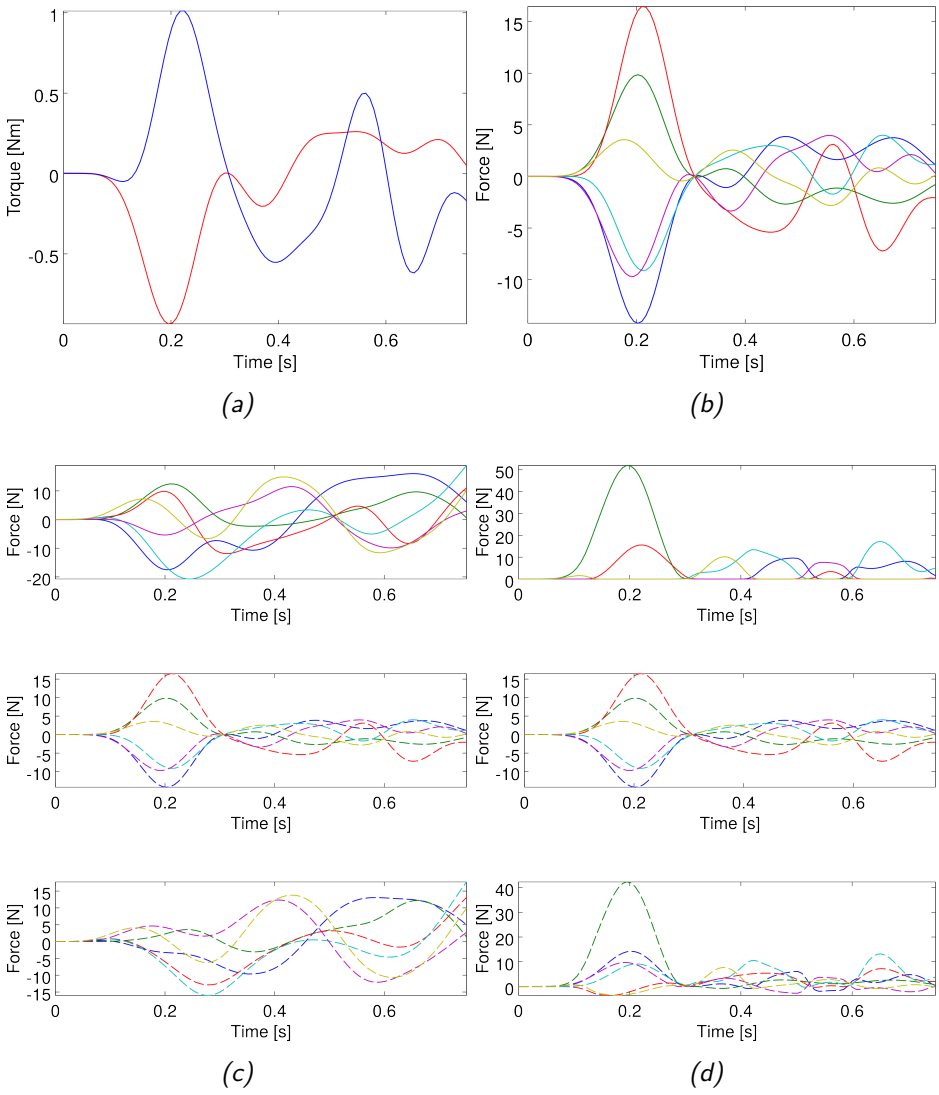


Figure 4.2: Effect of redundancy on the force model. (a) representative torque profile, where the red line refers to the elbow joint and the blue one to the shoulder joint. (b) shows a possible force contributing component that leads to the desired torque, where each color corresponds to a different muscle. (c) and (d) depict two different force profiles (top plot) that share the same contributing component (middle plot, also shown in (b)), and have different non-contributing components (bottom plots).

where $\boldsymbol{N} \in \mathbb{R}^{6\times4}$ is a matrix containing the basis of the nullspace of $\boldsymbol{L}$, and $\boldsymbol{w}_i(t) \in \mathbb{R}^4$ is the so called *nullshift* and can be freely changed without affecting the torque. A force $\boldsymbol{F}_i(t)$ leading to a torque $\boldsymbol{\tau}_i(t)$ can therefore be finally written as:

$$\boxed{\boldsymbol{F}_i(t) = \boldsymbol{F}_i^0(t) + \boldsymbol{F}_i^k(t) = \boldsymbol{L}^+\boldsymbol{\tau}_i(t) + \boldsymbol{N}\boldsymbol{w}_i(t)} \tag{4.5}$$

Since $\boldsymbol{N}$ contains a basis of $\ker(\boldsymbol{L})$, then $\boldsymbol{L}\boldsymbol{N}\boldsymbol{w}_i(t) = \boldsymbol{0}\ \forall \boldsymbol{w}_i(t)$ and the torque produced by $\boldsymbol{F}_i(t)$ does not depend on $\boldsymbol{w}_i(t)$.

Similarly a force synergy $\boldsymbol{\gamma}_j(t)$ can be found from its corresponding torque $\boldsymbol{\tau}_j(t)$ as follows:

$$\boldsymbol{\gamma}_j(t) = \boldsymbol{L}^+\boldsymbol{\tau}_j(t) + \boldsymbol{N}\boldsymbol{w}_j(t) \tag{4.6}$$

for some $\boldsymbol{w}_j(t)$. Generalizing to all synergies we can write:

$$\boxed{\boldsymbol{\Gamma}(t) = \boldsymbol{\Gamma}^0(t) + \boldsymbol{\Gamma}^k(t) = \boldsymbol{L}^+\mathbf{T}(t) + \boldsymbol{N}\boldsymbol{W}(t)} \tag{4.7}$$

where $\boldsymbol{\Gamma}(t) = [\boldsymbol{\gamma}_1(t) \dots \boldsymbol{\gamma}_{n_\gamma}(t)] \in \mathbb{R}^{6\times n_\gamma}$ is a formal matrix containing the synergies arranged in columns, $\mathbf{T}(t) = [\boldsymbol{\tau}_1(t) \dots \boldsymbol{\tau}_{n_\gamma}(t)] \in \mathbb{R}^{2\times n_\gamma}$ is the matrix with the corresponding torques, and $\boldsymbol{W}(t) = [\boldsymbol{w}_1(t) \dots \boldsymbol{w}_{n_\gamma}(t)] \in \mathbb{R}^{4\times n_\gamma}$ contains the nullshifts $\boldsymbol{w}_j(t)$ corresponding to each synergy. From now on, in the interest of readability, we will omit the independent time variable t in case this does not introduce ambiguities in interpreting formulas.

The concept of nullspace is important because all possible forces $\boldsymbol{F}_i$ leading to a particular torque $\boldsymbol{\tau}_i$ can be found by varying $\boldsymbol{w}_i$. Similarly, all possible synergies $\boldsymbol{\Gamma}$ leading to particular torque actuations $\mathbf{T}$ can be found by varying $\boldsymbol{W}$. The nullshift $\boldsymbol{W}$ and $\boldsymbol{w}_i$ can therefore be used to change the profiles of forces and synergies, in order to satisfy biological constraints and to obtain performance. Therefore, if one or more suitable torque-to-force relations exist (leading to forces satisfying biological constraints and to synergies capable of approximating them) these can be found across the possible $\boldsymbol{w}_i$ and $\boldsymbol{W}$.

Given a desired dataset of torques (that solve the tasks of interest), possible corresponding force datasets can be found by varying $\boldsymbol{w}_i$. Since this parameter affects the shapes of the force profiles, it will impact on the dimensionality of this force dataset. This means that more or less generators (i.e. synergies) are required to approximate it. It is therefore crucial to understand the effects of the nullshifts on the forces involved. In order to investigate this effect we have taken $n = 8$ torque that solve reaching tasks and have created $m = 100$ linear combinations of them using random mixing coefficients. In doing so we obtained a dataset $\{\boldsymbol{\tau}_i\}_{i=1}^m$ of torque actuations with dimensionality n. From this dataset we calculated $h = 100$ corresponding force datasets. We have done it in two different ways:

1. using a *fixed* nullshift $\boldsymbol{w}_j$, equal for each force, but different across the datasets. The force $\boldsymbol{F}_{ij}$ corresponding to the torque $\boldsymbol{\tau}_i$ in the dataset j is:

$$\boldsymbol{F}_{ij} = \boldsymbol{L}^+\boldsymbol{\tau}_i + \boldsymbol{N}\boldsymbol{w}_j \quad \text{for } i = 1 \dots m,\ j = 1 \dots h \tag{4.8}$$

2. using a *variable* nullshift $\boldsymbol{w}_{ij}$, different across both the forces and the datasets. The force $\boldsymbol{F}_{ij}$ corresponding to the torque $\boldsymbol{\tau}_i$ in the dataset j is:

$$\boldsymbol{F}_{ij} = \boldsymbol{L}^{+}\boldsymbol{\tau}_i + \boldsymbol{N}\boldsymbol{w}_{ij} \quad \text{for } i = 1 \dots m,\ j = 1 \dots h \tag{4.9}$$

In both cases, $\boldsymbol{w}_i$ and $\boldsymbol{w}_{ij}$ have been selected randomly. We have then evaluated the dimensionality of all the force datasets using SVD (Singular Value Decomposition). Singular values provide information on how much variability in the data they explain. The smaller the singular value, the less variability it explains. If one plots the singular values in decreasing order, the resulting plot goes to 0 when all the variability in the data is explained. The number of non-null singular values needed to explain most or all the variability gives an idea of the dataset dimensionality[1].

For each force dataset we plotted the singular values as described above. The singular values are normalized between 0 and 1 in order to make possible the comparison between quantities with different magnitude. Figure 4.3a shows the plots obtained from the datasets with fixed nullshifts $\boldsymbol{w}_j$. Figure 4.3b shows instead the plots obtained from the datasets with variable nullshifts $\boldsymbol{w}_{ij}$. The plots obtained from the different force datasets overlap almost perfectly in both cases. These two figures show that calculating each force with a different nullshift leads to a much higher dimensionality than when all forces are calculated using the same nullshift.

We then calculated the standard deviation of each singular value across the h datasets both in the case of fixed and variable nullshift. Our objective was to see if the particular value of $\boldsymbol{w}_j$ (respectively $\boldsymbol{w}_{ij}$) influenced the singular values, and therefore the dimensionality. The results are shown in Figure 4.3c (fixed nullshift) and in Figure 4.3d (variable nullshift). The histograms show that in the first case almost all singular values have the same standard deviation, while in the second case (of variable nullshift) the singular values present much more variability. This result suggests that when a force dataset is calculated with a fixed nullshift $\boldsymbol{w}_j$, not only the dimensionality is lower, but it does not depend on the particular $\boldsymbol{w}_j$ chosen. If the dimensionality would depend on the particular nullshift we would observe move variability in Figure 4.3c.

To conclude this short analysis, we compare the dimensionality of the force datasets with that of the original torque dataset which we know is

[1] In the rest of this manuscript we use SVD to provide an estimation of a dataset dimensionality. When we say that a dataset has lower dimensionality than another, we mean that, for the number of singular values we consider, the variability in the first dataset is lower and therefore it can be explained with a smaller number of generators.

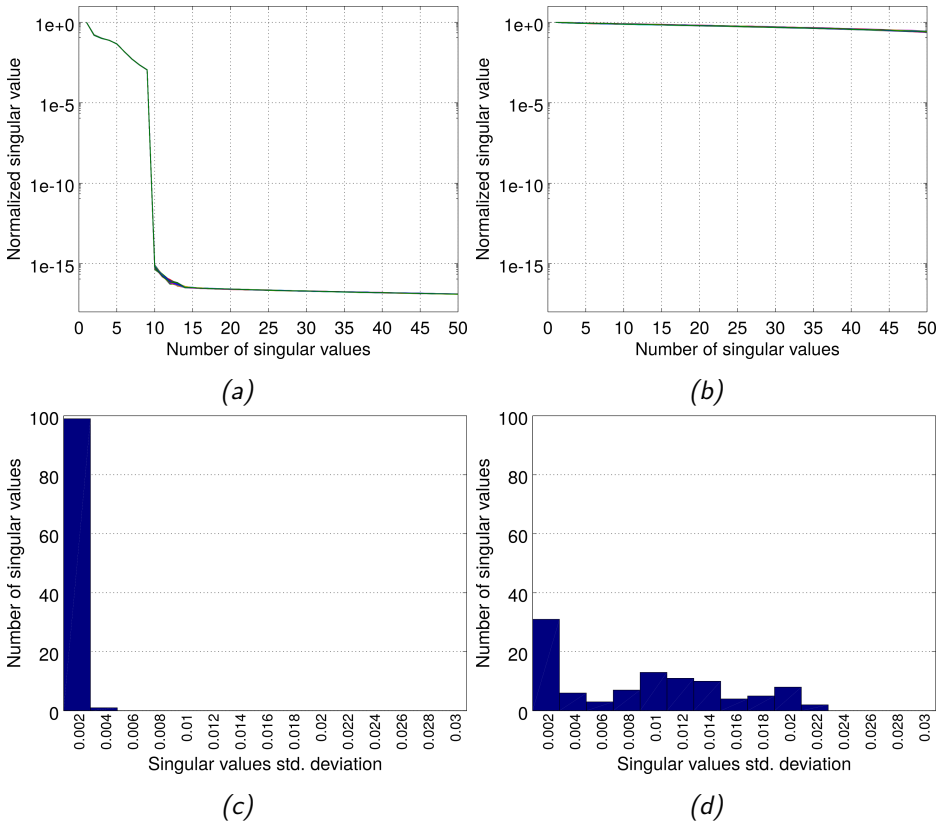


Figure 4.3: Dimensionality comparison between force datasets calculated from a torque dataset using nullshifts. (a) The $h = 100$ lines represent the normalized singular values of the h force datasets calculated each with a fixed nullshift $\boldsymbol{w}_j$, equal across all forces in the same dataset, but different across the datasets. (b) The $h = 100$ lines represent the normalized singular values of the h force datasets where each force is calculated with a variable nullshift $\boldsymbol{w}_{ij}$ different across forces and datasets. (c, d) Singular values standard deviation distribution of the force datasets in the case of fixed nullshift (c) and in the case of variable nullshift (d). Values on the x axis represent standard deviation bins. Values on the y axis represent how many singular values have a standard deviation falling inside a bin on the x axis.

$n = 8$. We calculated the mean singular values across the h force datasets in the case of fixed nullshift and variable nullshift. In other words we calculated the mean lines from Figures 4.3a and 4.3b and plotted them together with the singular values from the torque dataset. The result is shown in Figure 4.4. The torque dataset (blue line) has dimensionality $n = 8$ because the lines approaches 0 at the 9-th singular value. The force datasets calculated with fixed nullshifts (red line) have dimensionality $n + 1 = 9$, while the force datasets calculated with variable nullshifts (green line) have very high

dimensionality.

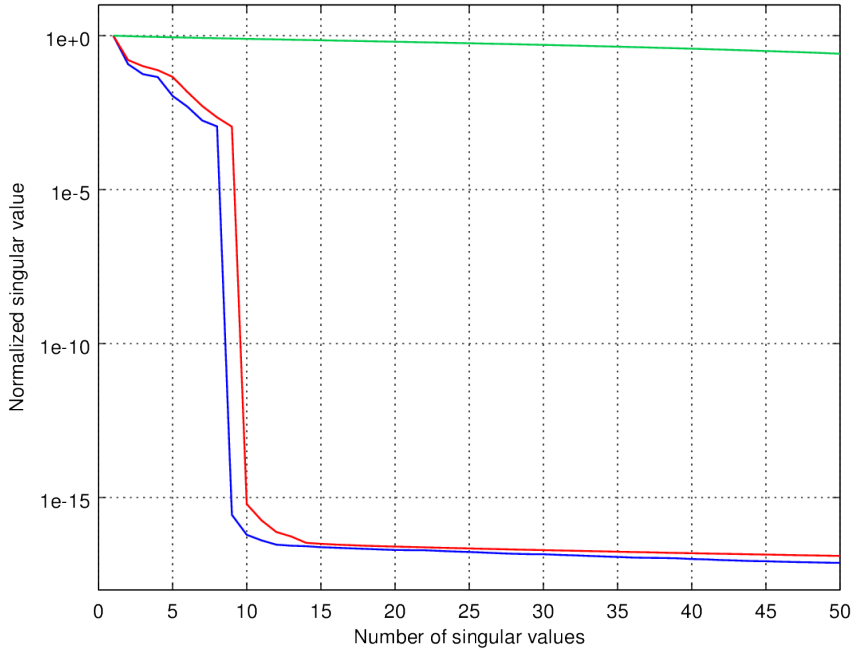


Figure 4.4: Dimensionality comparison between torque dataset (blue line), fixed nullshift force datasets (red line) and variable nullshift force datasets (green line). Singular values in the two latter cases are the mean of the singular values across all the datasets. The singular values are normalized between 0 and 1.

All these tests have been performed on a torque dataset obtained by linearly combining n torque generators. In order to ground these results to our problem we have taken 100 torques that solve desired reaching tasks as our torque dataset and have done the same analysis. The results are shown in Figure 4.5. The torque dataset (blue line) now has a much higher dimensionality than in the previous case. However, the dimensionality of the force dataset obtained with a fixed nullshift (red line) is still very similar to that of the torque dataset. On the other hand, the dimensionality of the variable nullshift force dataset (green line) is again much higher, meaning that much more generators would be needed to approximate it. These results suggest that to keep the forces dimensionality as low as possible we have to shift all forces with the same nullshift. This implies that in order to approximate this dataset we would need less synergies than if we use a variable nullshift.

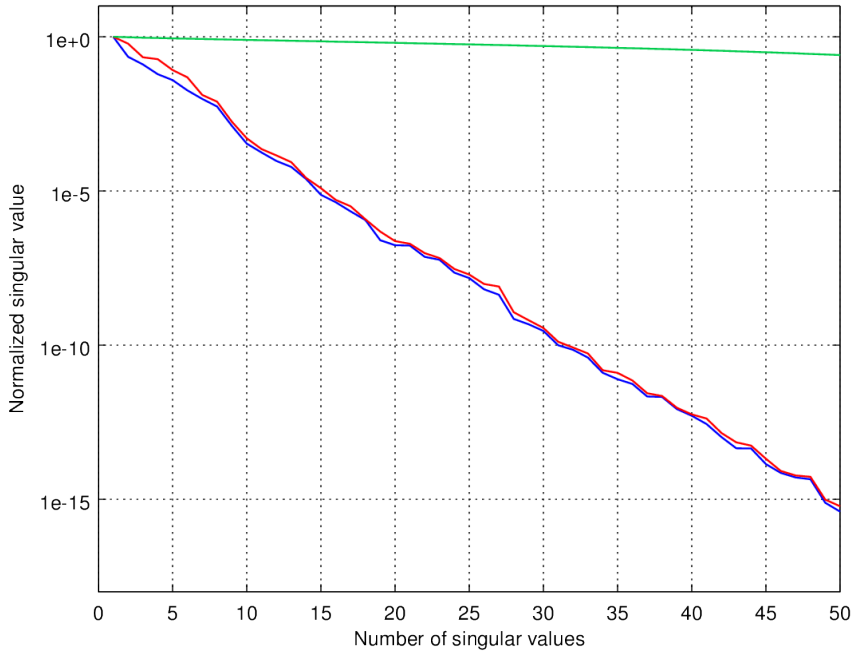


Figure 4.5: Dimensionality comparison between a torque dataset composed of 100 reaching task actuations (blue line), fixed nullshift force datasets (red line) and variable nullshift force datasets (green line). Singular values in the two latter cases are the mean of the singular values across all the datasets. The singular values are normalized between 0 and 1.

4.3 Problem formulation

In the previous sections we have shown that choosing the desired force dataset and the synergies corresponds to choosing $\boldsymbol{w}_i$ and $\boldsymbol{W}$ respectively, and that the dimensionality of the force dataset is similar to that of the torque dataset if we use the same nullshift for all the forces. In this section we formalize the general problem of finding $\boldsymbol{w}_i$ and $\boldsymbol{W}$.

Given a desired force $\boldsymbol{F}_i$ and the synergies $\boldsymbol{\Gamma}$, defined as in Eqs. (4.5) and (4.7) respectively, we want to approximate $\boldsymbol{F}_i$ with a linear combination of synergies, such that the resulting force is positive. In other words we want to find the force $\tilde{\boldsymbol{F}}_i = \boldsymbol{\Gamma}\boldsymbol{b}_i \geq \mathbf{0}$ which minimizes the norm of:

$$\boldsymbol{\varepsilon}_{\boldsymbol{F}} = \tilde{\boldsymbol{F}}_i - \boldsymbol{F}_i = \boldsymbol{\Gamma}\boldsymbol{b}_i - \boldsymbol{F}_i = (\boldsymbol{L}^{+}\mathbf{T} + \boldsymbol{N}\boldsymbol{W})\boldsymbol{b}_i - \boldsymbol{L}^{+}\boldsymbol{\tau}_i - \boldsymbol{N}\boldsymbol{w}_i \qquad (4.10)$$

where $\boldsymbol{b}_i = [b_{i1} \dots b_{in_\gamma}] \in \mathbb{R}^{n_\gamma}$ is a constant vector of mixing coefficients. This problem is ill-posed because its solution depends on the unknown null-

shifts $\boldsymbol{w}_i$ and $\boldsymbol{W}$ respectively, which however provide additional degrees of freedom in approximating $\boldsymbol{F}_i$. This problem can be formulated as follows:

$$\begin{aligned}[\boldsymbol{b}_i, \boldsymbol{w}_i, \boldsymbol{W}] &= \underset{\boldsymbol{b}_i, \boldsymbol{w}_i, \boldsymbol{W}}{\operatorname{argmin}} \|\boldsymbol{\Gamma}\boldsymbol{b}_i - \boldsymbol{F}_i\| \\ &= \underset{\boldsymbol{b}_i, \boldsymbol{w}_i, \boldsymbol{W}}{\operatorname{argmin}} \|(\boldsymbol{L}^+\mathbf{T} + \boldsymbol{N}\boldsymbol{W})\boldsymbol{b}_i - \boldsymbol{L}^+\boldsymbol{\tau}_i - \boldsymbol{N}\boldsymbol{w}_i\| \\ &\quad \text{s.t.} \quad (\boldsymbol{L}^+\mathbf{T} + \boldsymbol{N}\boldsymbol{W})\boldsymbol{b}_i \geq \mathbf{0}\end{aligned} \tag{4.11}$$

where the norm $\|\cdot\|$ operator represents, when applied to a function $\boldsymbol{f}(t)$, the generalization of the Euclidean norm of a vector and is calculated as follows:

$$\|\boldsymbol{f}(t)\| = \sqrt{\int_{t\in T} |\boldsymbol{f}(t)|^2 \mathrm{d}t}$$

Problem (4.11) is very complex and computationally intensive, furthermore it may lead to different $\boldsymbol{W}$ for each force to be approximated. This means that the synergies would be task-dependent, which is not desiderable and conflicts with the basic assumption of the muscle synergy hypothesis. For the moment, let's assume that $\boldsymbol{W}$ is given. Under this assumption $\boldsymbol{w}_i$ can be written as a function of $\boldsymbol{b}_i$. Solving the system of linear equations (4.10) for $\boldsymbol{w}_i$, all possible solutions minimizing $\boldsymbol{\varepsilon}_{\boldsymbol{F}}$ can be expressed as follows:

$$\boldsymbol{w}_i = \boldsymbol{N}^+(\boldsymbol{L}^+\mathbf{T}\boldsymbol{b}_i + \boldsymbol{N}\boldsymbol{W}\boldsymbol{b}_i - \boldsymbol{L}^+\boldsymbol{\tau}_i), \tag{4.12}$$

and since $\boldsymbol{N}^+\boldsymbol{N} = \boldsymbol{I}$ and $\boldsymbol{N}^+\boldsymbol{L}^+ = \mathbf{0}$ we obtain the fundamental relation:

$$\boldsymbol{w}_i = \boldsymbol{W}\boldsymbol{b}_i. \tag{4.13}$$

Substituting (4.13) in (4.11), the final expression of the general problem becomes:

$$\begin{aligned}\boldsymbol{b}_i &= \underset{\boldsymbol{b}_i}{\operatorname{argmin}} \|\boldsymbol{L}^+\mathbf{T}\boldsymbol{b}_i - \boldsymbol{L}^+\boldsymbol{\tau}_i\| \\ &\quad \text{s.t.} \quad (\boldsymbol{L}^+\mathbf{T} + \boldsymbol{N}\boldsymbol{W})\boldsymbol{b}_i \geq \mathbf{0}\end{aligned} \tag{4.14}$$

where the cost function in $\|\cdot\|$ does not depend anymore on $\boldsymbol{w}_i$ and $\boldsymbol{W}$, and the problem only consists in finding the appropriate mixing coefficients $\boldsymbol{b}_i$ which minimizes the error. However, the particular value of the synergies nullshift $\boldsymbol{W}$ affects the constraint of the optimization, thus reducing the space of the possible solutions. The choice of the right $\boldsymbol{W}$ becomes therefore crucial. In the next section we consider different possible models for $\boldsymbol{W}$.

4.4 Identifying synergies

In the previous section we have seen that, once the synergies are defined, a desired actuation can be approximated by solving the problem (4.14), where the constraint depends on the value of the synergy nullshift $\boldsymbol{W}$. In this section we consider some possible choices for the value of $\boldsymbol{W}$.

The first question that must be addressed is how to choose the individual nullshifts of the synergies, corresponding to the columns of $\boldsymbol{W}$. In order to strive for better performance in approximating the desired forces one should choose the nullshift to make the problem (4.14) as less constrained as possible, in order to make the space containing all feasible $\boldsymbol{b}_i$ as large as possible. This however would be very complex to achieve and depends on the particular prototasks which are not known a-priori. Furthermore, in section 4.2 we have shown that in order to reduce the dimensionality of the force dataset, we should use the same $\boldsymbol{w}_i$ for all desired forces. Such a transformation is an affine map between the torque space and the force space. Similarly we assume that shifting all synergies by a single $\boldsymbol{w}^*$ might be beneficial in terms of the capability to approximate a force dataset[2]. In this thesis therefore we considered a model consisting in all synergies shifted by a nullshift $\boldsymbol{w}^*$:

$$\boldsymbol{W} = [\boldsymbol{w}^* \dots \boldsymbol{w}^*] \tag{4.15}$$

Later, in section 4.4.3, we will show that this assumption is in accordance with the results on the dimensionality reported in section 4.2.

Another important question to take into consideration, is whether synergies represent "valid" actuations or not, that is, whether they fulfill biological constraints (i.e. $\mathbf{0} \leq \boldsymbol{\Gamma} \leq \boldsymbol{F}_{max}$), or if they represent instead abstract generators which, when linearly combined, lead to valid forces. In neuroscience there is no agreement on this matter. We therefore considered both cases: synergies that fulfill biological constraints (in particular, positivity), and synergies that don't.

From the above considerations we identified three possible models:

A No nullshifts are applied to the synergies ($\boldsymbol{w}^* = \mathbf{0}$);

B The same nullshift is applied to all synergies, without constraining them to be positive ($\boldsymbol{w}^* \neq \mathbf{0}$);

C The same nullshift is applied to all synergies such that they become positive ($\boldsymbol{w}^* \neq \mathbf{0} : \boldsymbol{L}^+\mathbf{T} + \boldsymbol{N}\boldsymbol{W} > \mathbf{0}$).

[2] Additional work is required to investigate if this assumption is theoretically grounded.

4.4.1 Model A: No nullshift

In this model the synergy nullshift is defined as:

$$\boldsymbol{W} = [\mathbf{0} \dots \mathbf{0}] \tag{4.16}$$

From Eq. (4.7) the force synergies are calculated from the corresponding torque actuations as $\boldsymbol{\Gamma} = \boldsymbol{L}^{+}\mathbf{T}$. Given that this expression returns the minimum norm solution, the synergies are in general both positive and negative. Since $\boldsymbol{W} = \mathbf{0}$, from Eqs. (4.13) and (4.5) it follows that for any desired force $\boldsymbol{F}_i$ to be approximated:

$$\boldsymbol{w}_i = \mathbf{0} \Rightarrow \boldsymbol{F}_i = \boldsymbol{L}^{+}\boldsymbol{\tau}_i \tag{4.17}$$

For the same reasons already discussed above for the synergies $\boldsymbol{\Gamma}$, also the desired force $\boldsymbol{F}_i$ is not constrained to be positive. At least two considerations can be done about the approximating force $\tilde{\boldsymbol{F}}_i$ found through (4.14). First, the difference $\boldsymbol{\varepsilon}_F$ between $\tilde{\boldsymbol{F}}_i$ (which is positive) and the desired force $\boldsymbol{F}_i$ that instead is both positive and negative (see above) is probably not negligible. Second, generating a positive force ($\tilde{\boldsymbol{F}}_i$) with unconstrained synergies is only possible if the synergies have very particular profiles. Given that the synergies do not have any particular form (they are minimum norm solutions of the torque actuations $\mathbf{T}$) we expect that the problem (4.14) is either infeasible, or leads to very high projection error. In this model in fact the mixing coefficient $\boldsymbol{b}_i$ found in (4.14) tend to zero in almost every case, because this is the only value which always satisfies the constraints. We have verified this observation by trying to approximate over 800 desired forces that correspond to reaching tasks homogeneously distributed in the operational space. Figure 4.6 shows that the components b_{ij} in $\boldsymbol{b}_i = [b_{i1} \dots b_{in_\gamma}]$ tend to zero almost for every task i. Figure 4.7 shows the mean relative projection error (left) and the mean forward dynamics error (right) plotted against the number of synergies. As expected this model is not able to approximate the desired forces.

4.4.2 Model B: Same nullshift without positivity constraints

The nullshift is defined as:

$$\boldsymbol{W} = [\boldsymbol{w}^* \dots \boldsymbol{w}^*] \tag{4.18}$$

and the synergies are not subject to any positivity constraint. From Eq. (4.13), in order to minimize the projection error, the nullshifts of the desired

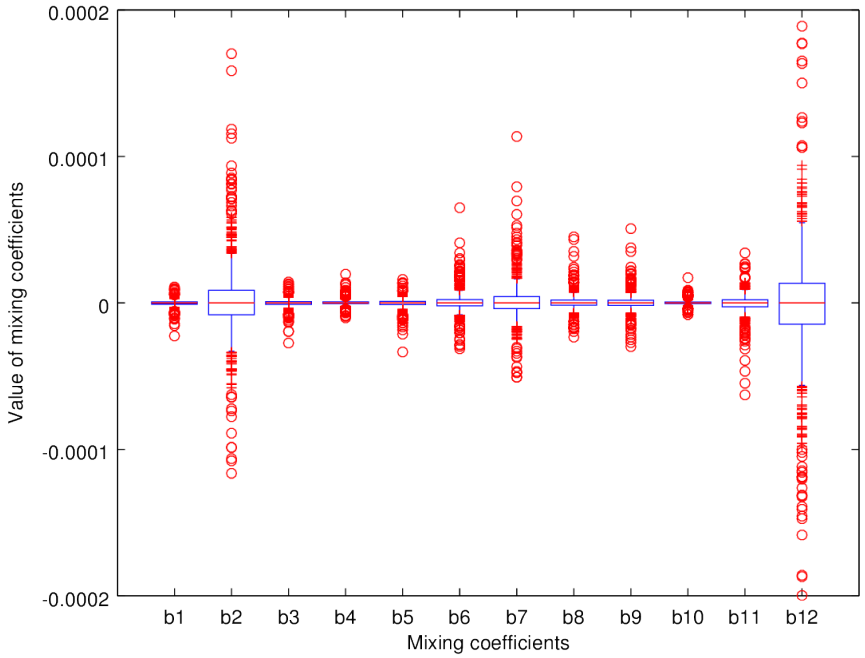
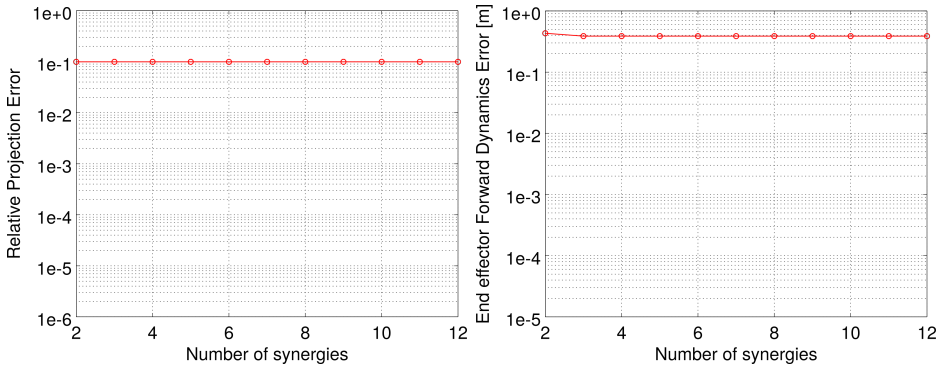


Figure 4.6: Distribution of the mixing coefficient components for 12 non-constrained synergies with $\boldsymbol{W} = \boldsymbol{0}$. Each boxplot represents the distribution of the mixing coefficient across 800 tasks. All mixing coefficients are very small, meaning that for each task i $\boldsymbol{b}_i \to \boldsymbol{0}$.



(a) *(b)*

Figure 4.7: Relative projection error vs. number of synergies (a) and forward dynamics error vs. number of synergies (b) for the model with non-constrained synergies with $\boldsymbol{W} = \boldsymbol{0}$.

forces have to be:

$$\boldsymbol{w}_i = \boldsymbol{W}\boldsymbol{b}_i = \boldsymbol{w}^* \sum_{j=1}^{n_\gamma} \boldsymbol{b}_{ij} \Rightarrow \boldsymbol{F}_i = \boldsymbol{L}^+ \boldsymbol{\tau}_i + \boldsymbol{N}\boldsymbol{w}^* \sum_{j=1}^{n_\gamma} \boldsymbol{b}_{ij} \tag{4.19}$$

However, this "optimal" $\boldsymbol{w}_i$ does not guarantee that the desired force $\boldsymbol{F}_i$ is positive. Hence, the same considerations already done for the previous model can be done also in this case.

4.4.3 Model C: Same nullshift with positivity constraints

In this model the nullshift is defined as:

$$\boldsymbol{W} = [\boldsymbol{w}^* \dots \boldsymbol{w}^*] \quad \text{such that} \quad \boldsymbol{\Gamma} \geq \boldsymbol{0} \tag{4.20}$$

From Eq. (4.13) and (4.5) the nullshifts that reduce the projection error and the corresponding forces are:

$$\boldsymbol{w}_i = \boldsymbol{W}\boldsymbol{b}_i = \boldsymbol{w}^* \sum_{j=1}^{n_\gamma} \boldsymbol{b}_{ij} \Rightarrow \boldsymbol{F}_i = \boldsymbol{L}^+\boldsymbol{\tau}_i + \boldsymbol{N}\boldsymbol{w}^* \sum_{j=1}^{n_\gamma} \boldsymbol{b}_{ij} \tag{4.21}$$

While there is no guarantee that the desired forces $\boldsymbol{F}_i$ are positive, we can be sure that the problem (4.14) is feasible because $\boldsymbol{\Gamma} \geq \boldsymbol{0}$, and therefore its constraint (i.e. $\boldsymbol{\Gamma}\boldsymbol{b}_i \geq \boldsymbol{0}$) can be satisfied for some $\boldsymbol{b}_i \neq \boldsymbol{0}$. The question is how to find a suitable $\boldsymbol{w}^*$. A possible strategy is to define an optimization problem that minimizes an appropriate cost function $\boldsymbol{g}(\cdot)$, and to impose the constraint $\boldsymbol{\Gamma} \geq \boldsymbol{0}$:

$$\begin{aligned} \boldsymbol{w}^* = \underset{\boldsymbol{w}}{\operatorname{argmin}}\, \boldsymbol{g}(\boldsymbol{w}) \\ \text{s.t.} \quad \boldsymbol{\Gamma} = \boldsymbol{L}^+\mathbf{T} + \boldsymbol{N}\boldsymbol{W} \geq \boldsymbol{0} \end{aligned} \tag{4.22}$$

As an example we calculated a $\boldsymbol{w}^*$ for a set of 8 predefined prototasks, using 4 different cost functions, thus leading to 4 different sets of synergies. The particular cost functions we have used are not relevant for the following discussion (they are briefly discussed in Figure 4.8). For each $\boldsymbol{w}^*$ we approximated over 800 desired forces (corresponding to reaching tasks homogeneously distributed in the end effector space) as linear combinations of synergies (i.e. $\boldsymbol{\Gamma}\boldsymbol{b}_i$), with mixing coefficients calculated as in Eq. (4.14). The nullshift $\boldsymbol{w}_i$ of each force was calculated by means of Eq. (4.13). We surprisingly found that, in all cases, all $\boldsymbol{w}_i$ tended to the particular $\boldsymbol{w}^*$ used for the synergies, that is:

$$\forall \boldsymbol{F}_i = \boldsymbol{L}^+\boldsymbol{\tau}_i + \boldsymbol{N}\boldsymbol{w}_i,\ \boldsymbol{w}_i \to \boldsymbol{w}^* \tag{4.23}$$

Results for the 4 representative $\boldsymbol{w}^*$ are shown in Figure 4.8, where the red thick line corresponds to $\boldsymbol{w}^*$, and the colored lines represent the $\boldsymbol{w}_i$. This result is in accordance with the results obtained in the dimensionality

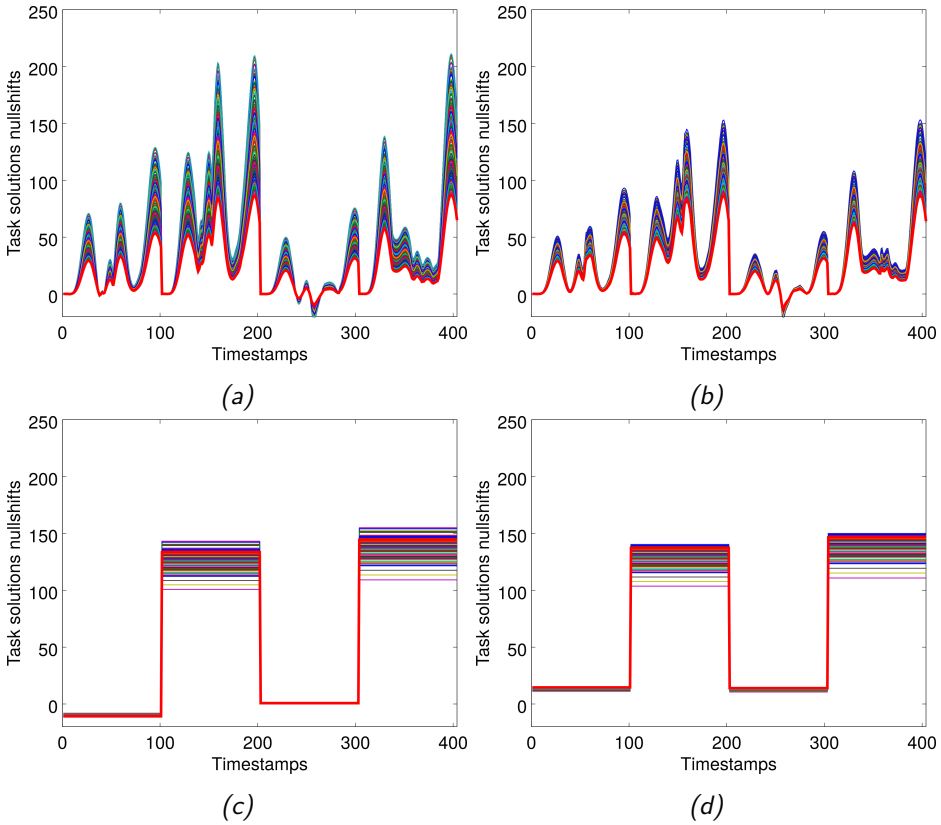


Figure 4.8: Examples of force nullshifts $\boldsymbol{w}_i$ calculated using Eqs. (4.14) and (4.13) for 4 different $\boldsymbol{w}^*$. The colored lines are the nullshifts $\boldsymbol{w}_i$ associated to 800 desired forces. The red thick line represents the $\boldsymbol{w}^*$ used for the synergies. In each case all $\boldsymbol{w}_i$ converge to $\boldsymbol{w}^*$. Note that $\boldsymbol{w}^*, \boldsymbol{w}_i \in \mathbb{R}^4$ and in order to plot them their dimensions have been concatenated: the first dimension goes from timestamp 1 to 100, the second from 101 to 200 and so on. (a) nullshift found by linearly minimizing the sum of the total shift $\boldsymbol{N}\boldsymbol{w}^*$. (b) nullshift found by minimizing the norm of the total shift $\boldsymbol{N}\boldsymbol{w}^*$. (c) nullshift leading to $\boldsymbol{N}\boldsymbol{w}^* = [k \dots k]^T$ with k the minimum constant value necessary to make positive all forces in all synergies. (d) nullshift leading to $\boldsymbol{N}\boldsymbol{w}^* = [k_1 \dots k_6]^T$ with k_i the minimum constant value necessary to make positive the force i in all synergies.

analysis in section 4.2. There we found that a single nullshift preserves the dimensionality of the force dataset. Now we are showing that choosing a fixed nullshift for the synergies, and minimizing the projection error leads to nullshifts of the desired forces which approach that of the synergies, resulting in a single fixed nullshift.

In the next sections we introduce the results from the reduction procedure when using a fixed nullshift $\boldsymbol{w}^*$ to calculate the force synergies, and show that the obtained performance are comparable to those of the torque

model.

4.5 Reduction

As explained in section 2.3 the reduction consists in iteratively finding kinematic solutions to the prototasks, and in computing the corresponding actuations by means of inverse dynamics. These actuations will be used as synergies. In the force model, this computation involves using the torque-to-force relation to translate the torque that solves a prototask into a force. Such a relation includes the $\boldsymbol{w}^*$ discussed above. A possible method to find $\boldsymbol{w}^*$ could be to minimize a cost function with the constraint that the synergies must be positive, as described in section 4.4.3. This approach has the problem that it assumes an a-priori knowledge of the synergies $\boldsymbol{\Gamma}$. On the other hand, synergies are computed iteratively via the reduction procedure. Moreover, the following considerations hold. We have shown that the nullshift $\boldsymbol{w}_i$ of the desired forces converges to $\boldsymbol{w}^*$. Thus it seems to be that, by minimizing the error $\boldsymbol{\varepsilon_F}$ as in Eq. (4.14), the model naturally converges to one where all synergies and all desired forces are shifted by the same nullshift. From Figure 4.8, however, it can be noticed that $\boldsymbol{w}_i$ and $\boldsymbol{w}^*$ are not exactly equal, though very similar. We believe that the distance between the particular $\boldsymbol{w}_i$ and the synergy nullshift $\boldsymbol{w}^*$ depends on whether $\boldsymbol{w}^*$ is able to positivize the desired forces. If this is the case (i.e. $\boldsymbol{F}_i = \boldsymbol{L}^+\boldsymbol{\tau}_i + \boldsymbol{N}\boldsymbol{w}^* \geq \boldsymbol{0}$) then the difference between $\boldsymbol{w}_i$ and $\boldsymbol{w}^*$ will be zero, and the error $\boldsymbol{\varepsilon_F}$ in Eq. (4.10) might become smaller. To address these issues, a possible solution is to search a nullshift which positivizes a set of representative forces instead of only positivizing the synergies as in Eq. (4.22). The idea is that such a $\boldsymbol{w}^*$ encapsulates informations on the desired forces. We picked therefore 200 random reaching tasks and used a linear programming problem in order to find $\boldsymbol{w}^*$:

$$\begin{aligned} \boldsymbol{w}^*(t) = \underset{\boldsymbol{w}(t)}{\operatorname{argmin}} & \sum_t \boldsymbol{N}\boldsymbol{w}(t) \\ \text{s.t.} \quad & \boldsymbol{L}^+\boldsymbol{\tau}_i(t) + \boldsymbol{N}\boldsymbol{w}(t) \geq \boldsymbol{0} \ \text{ for } \ i = 1..200 \end{aligned} \tag{4.24}$$

We then used this $\boldsymbol{w}^*$ as a first guess to calculate the synergies via the iterative process of the reduction. At iteration j of the reduction procedure, a new synergy $\boldsymbol{\gamma}_j$ is calculated from the torque actuation $\boldsymbol{\tau}_j$ that solves the new prototask as follows:

$$\boldsymbol{\gamma}_j = \boldsymbol{L}^+\boldsymbol{\tau}_j + \boldsymbol{N}\boldsymbol{w}^* \tag{4.25}$$

The $\boldsymbol{w}^*$ however, being calculated only on a set of representative tasks as in Eq. (4.24), does not guarantee the positivity of the new synergy. Therefore at each iteration j we adjusted the current $\boldsymbol{w}^*$ in order to satisfy the positivity constraints, and we recalculated all synergies with the new $\boldsymbol{w}^*$. Such an adjustment is performed by means of the following optimization:

$$\begin{aligned} \boldsymbol{w}^* = \boldsymbol{w}_j^* = \underset{\boldsymbol{w}_j}{\operatorname{argmin}} \|\boldsymbol{w}_j - \boldsymbol{w}_{j-1}^*\| \\ \text{s.t.} \quad \boldsymbol{\Gamma} \geq \mathbf{0} \end{aligned} \tag{4.26}$$

The initial guess of $\boldsymbol{w}^*$ at iteration 1 is calculated from Eq. (4.24). The initial guess at iteration $j > 1$ was instead $\boldsymbol{w}_{j-1}^*$. Note that, given the argument of the minimization, the change of $\boldsymbol{w}^*$ between one iteration and the other should be very small. If at iteration j all synergies are already positive, the value of the nullshift is not changed. The $\boldsymbol{w}^*$ found at the last iteration renders all synergies positive, and it still encapsulates information from the tasks used to calculate its initial guess.

The result of the reduction procedure on the force model, iterated 12 times, is shown in Figure 4.9. Panels are ordered in row-major order, each corresponding to one iteration. The color code indicates the projection error in solving the 800 reaching tasks distributed in the reachable area. Table 4.1 shows the prototask coordinates for the force model, expressed in elbow and shoulder angular position. We started the reduction process with the same 2 prototasks used for the torque model. At each iteration, if the highest error was localized similarly as in the torque model we simply used the same prototask, otherwise we chose a new one. The two models started to diverge after the 8th prototask.

4.6 Performance

We compared the performance of the force model to those obtained for the torque model (presented in section 3). Figure 4.10 shows the mean relative projection error plotted against the number of synergies. The blue line represents the torque model, the red line the force model. Approximation performance are very similar, although the force model has slightly smaller errors. The forward dynamics error plotted against the number of synergies is shown in Figure 4.11. The blue line represents the torque model, while the red line the force model. From this plot it is evident that the average performances are practically the same and scale similarly by adding new synergies.

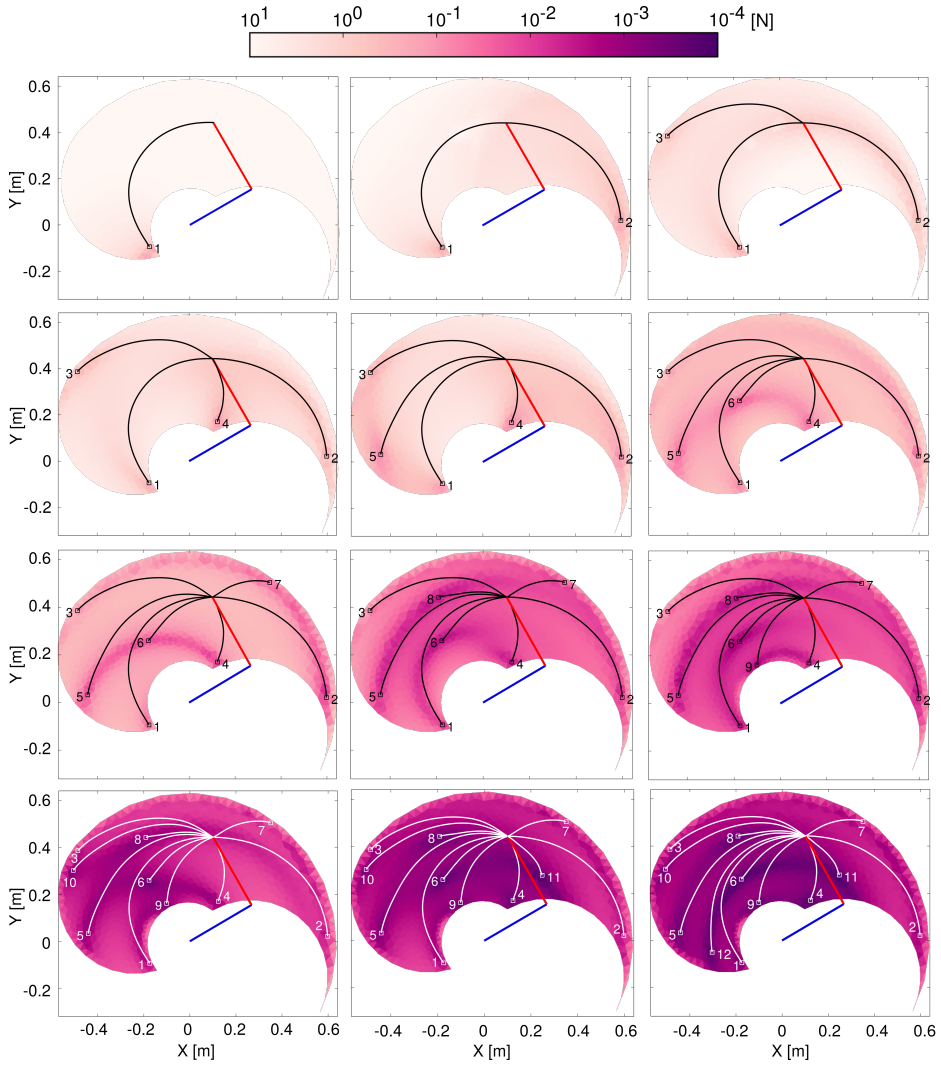


Figure 4.9: Results of the reduction procedure for the force actuated kinematic chain. Each panel corresponds to a number of synergies. Each point of the operational space is colored depending on the performance of the synergies in approximating a reaching task solution with final position in that point. Bright areas correspond to high projection errors while dark areas correspond to small projection errors as depicted in the color bar above.

Despite the higher complexity of the musculoskeletal model (due to redundancies and biological constraints) we were able to perform reaching tasks in the entire end effector space with similar performance as those obtained on the non-redundant and non-constrained torque model. These result suggests that muscle redundancy does not necessary increases the number of synergies required to obtain a certain performance.

Table 4.1: Prototasks positions in the force model, expressed as elbow and shoulder joint angles.

#	q_e [rad]	q_s [rad]
1	2.5183	2.2611
2	0.7411	-0.3497
3	0.5114	2.2047
4	2.4789	-0.4042
5	1.6219	2.2180
6	2.1195	1.0429
7	0.5884	0.6575
8	1.4434	1.2242
9	2.5437	0.7392
10	0.8163	2.1790
11	1.9065	-0.1651
12	2.1451	2.1641

These results where obtained by using the same nullshift $\boldsymbol{w}^*$ for all synergies. The nullshifts $\boldsymbol{w}_i$ of the desired forces $\boldsymbol{F}_i$, were found with Eqs. (4.14) and (4.13). Figure 4.12 depicts $\boldsymbol{w}^*$ (red thick line) and the individual $\boldsymbol{w}_i$ (colored lines) of the desired forces. As hypothesized in the previous section, the distance between $\boldsymbol{w}_i$ and $\boldsymbol{w}^*$ is very small, and the reason is that we calculated $\boldsymbol{w}^*$ from a large set of representative tasks, and not only from the synergies. As a result $\boldsymbol{w}^*$ is able to positivize lot of desired forces, for which $\boldsymbol{w}_i = \boldsymbol{w}^*$. Results suggest that a fixed nullshift not only preserves dimensionality, but also represents a good strategy to obtain good approximation and task performance.

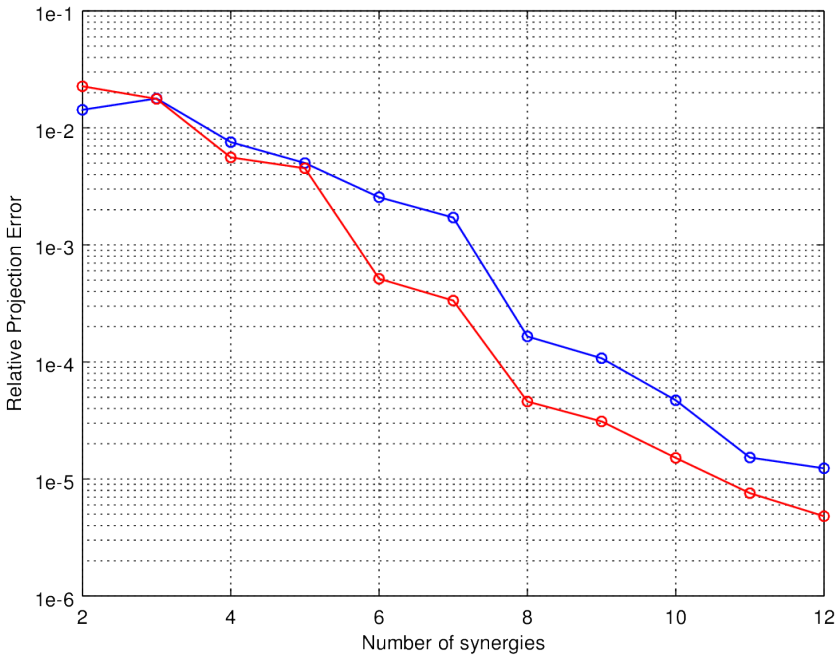


Figure 4.10: Relative projection error vs. number of synergies for the force model (red line) compared to the torque model (blue line).

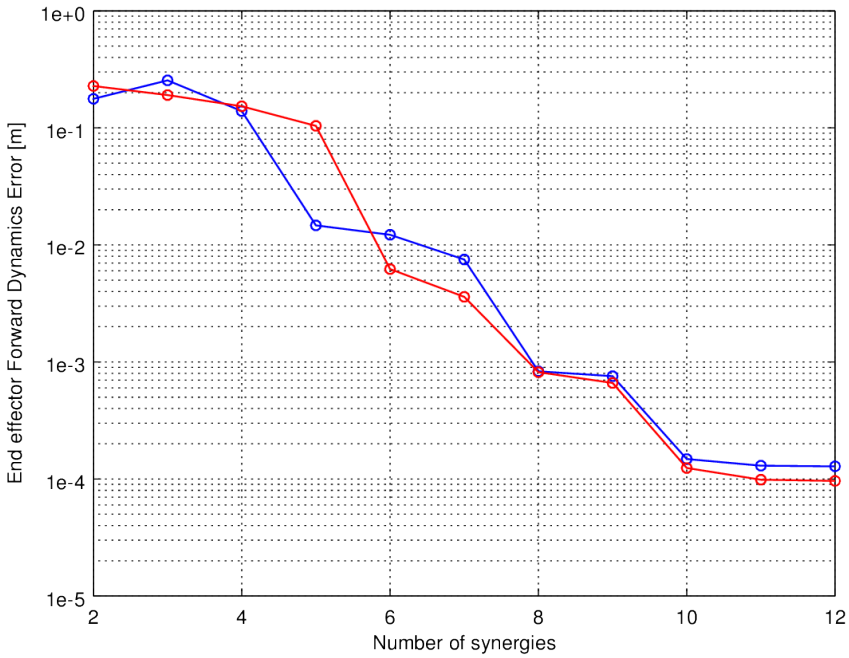


Figure 4.11: End effector forward dynamics error vs. number of synergies for the force model (red line) compared to the torque model (blue line).

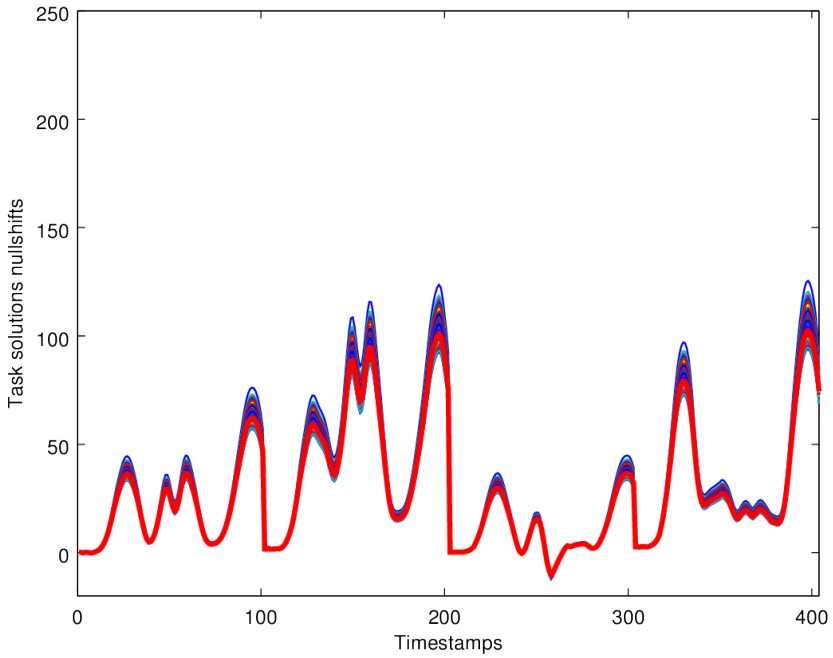


Figure 4.12: The optimal nullshift of the desired forces (colored lines) converge to that of the synergies (red thick line).

Chapter 5

Results: Activation Model

In this chapter we present the results obtained in the activation model, thus actuating the kinematic chain by means of nonlinear muscle models. We then formalize the inverse dynamics of the model in a way which enables the use of a nullshift also in the activation space. Finally, we report the results obtained in the activation model, and show that nonlinearities may increase the number of synergies required to obtain a certain performance, compared to the torque and force model.

5.1 Torque to activation relation

In the activation model forces, applied to the links of the kinematic chain, are produced by means of muscle models that are controlled in activation (see section 2.2.5). Like in the force model, in order to apply the DRD method we need to define the inverse dynamics of the system: given a trajectory in kinematic space, we need to find the activation which leads to that trajectory. The problems we face are therefore similar to those described in the previous chapter, and all considerations about redundancy and biological constraints still hold. In the following we ground these considerations to the activation model and describe the nonlinearities introduced by muscles.

The kinematic chain actuated in activation has more muscles than joints, thus it is a redundant system. As a result different activations can lead to the same torque, and therefore to the same trajectory. Note, however, that there is no additional redundancy in the dynamics of the individual muscles. Ones their lengths and contraction velocities are known, the generated force can be univocally determined from the activation. Lengths and contraction velocities can in turn be univocally determined from the state of the kine-

matic chain[1]. The redundancy therefore arises solely from the fact that the number of forces (generated by muscles) is higher than the number of joints. As we have seen, these forces must satisfy the positivity constraint and they cannot exceed a maximal value. Activation is defined in the range $[0, 1]$ and scales the output of the CE (Contractile Element) which generates the active force (see section 2.2.5). The problem, therefore, consists in finding a *torque-to-activation* relation $\boldsymbol{g}(\cdot)$, from torque space $\mathrm{T} \subseteq \mathbb{R}^2$ to activation space $\mathcal{M} \subseteq \mathbb{R}^6$, such that its inverse corresponds to relation (2.48):

$$\begin{aligned} \boldsymbol{g} : \mathrm{T} \to \mathcal{M} \quad : \\ \boldsymbol{\tau}(t) = \boldsymbol{LC}(\boldsymbol{l}, \dot{\boldsymbol{l}})\boldsymbol{M}(t) + \boldsymbol{LP}(\boldsymbol{l}) = \boldsymbol{LC}(\boldsymbol{l}, \dot{\boldsymbol{l}})\boldsymbol{g}(\boldsymbol{\tau}(t)) + \boldsymbol{LP}(\boldsymbol{l}) \\ \wedge \\ \boldsymbol{g}(\boldsymbol{\tau}(t)) \in [\boldsymbol{0}, \boldsymbol{1}] \quad \forall \boldsymbol{\tau}(t) \in \mathrm{T} \end{aligned} \tag{5.1}$$

where $\boldsymbol{\tau}(t)$ is the torque, $\boldsymbol{M}(t)$ the muscle activation and $\boldsymbol{L}$ the lever arm. $\boldsymbol{C}(\boldsymbol{l}, \dot{\boldsymbol{l}})$ and $\boldsymbol{P}(\boldsymbol{l})$ where introduced in section 2.2.5 and represent the state-dependent muscle model. $\boldsymbol{C}(\boldsymbol{l}, \dot{\boldsymbol{l}})$ relates muscle activation to the active force generated by the CE, while $\boldsymbol{P}(\boldsymbol{l})$ is the passive force produced when the muscle is extended over a certain length.

From now on, in the interest of readability, we will omit the time dependence t and the state variables $\boldsymbol{l}$ and $\dot{\boldsymbol{l}}$, unless this introduces ambiguities in interpreting formulas.

5.1.1 Nonlinearities

The activation model features nonlinearities and state-dependencies that are introduced by the nonlinear CE force-length and force-velocity relations (represented by the $\boldsymbol{C}$ term), and by the nonlinear PE force-length relation (represented by the $\boldsymbol{P}$ term). Figures 2.7 and 2.8 show the extent to which muscle dynamics is nonlinear and state-dependent.

The mechanisms underlying activation and force models are schematically represented in Figure 5.1. In the force model (Figure 5.1a) the force $\boldsymbol{F}$ acts on the joints through the fixed lever arm $\boldsymbol{L}$, producing the torque $\boldsymbol{\tau}$. In the activation model (Figure 5.1b) the relation between the total force $\boldsymbol{F}$ and the torque $\boldsymbol{\tau}$ is still linear, but the force $\boldsymbol{F}$ depends nonlinearly on the activation $\boldsymbol{M}$ and on the state of the kinematic chain $(\boldsymbol{q}, \dot{\boldsymbol{q}})$. Therefore,

[1]The muscle model is "rigid" because there is no Hill-like serial element modeling tendon elasticity. Muscle lengths and velocities are determined through Eqs. (2.42) and (2.43). Note that, from the definition of $\boldsymbol{L}$ given in Eq. (2.30), the kernel of $\boldsymbol{L}^T$ is empty, therefore the relation between muscle kinematics and kinematic chain state is one-to-one.

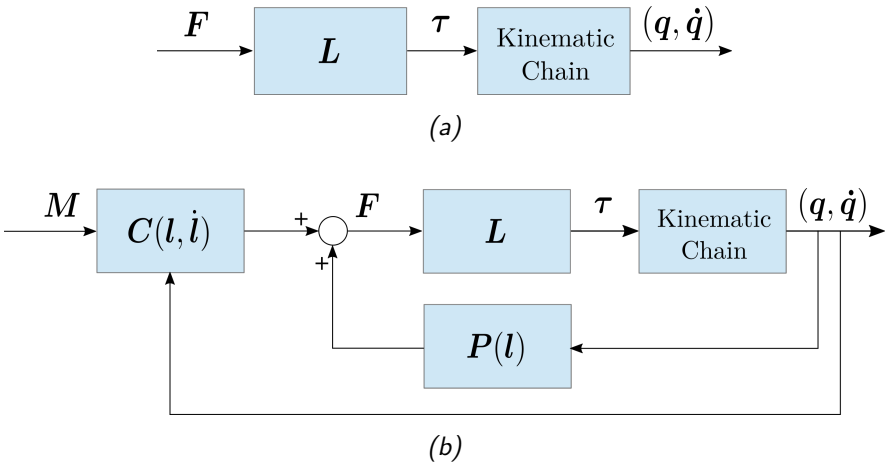


Figure 5.1: Comparison between force and activation model dynamics. (a) In the force model the force $\boldsymbol{F}$ acts on the joints producing a torque. The lever arm relation between force $\boldsymbol{F}$ and torque $\boldsymbol{\tau}$ is linear. (b) In the activation model the activation $\boldsymbol{M}$ produces a force $\boldsymbol{F}$ which nonlinearly depends on the kinematic state $(\boldsymbol{q}, \dot{\boldsymbol{q}})$. This force then linearly produces a torque $\boldsymbol{\tau}$ through the lever arm relation.

in order to find an activation corresponding to a given torque we have to (1) find the corresponding force, facing the redundancy problem, (2) decompose this force in its active and passive components (associated to $\boldsymbol{C}$ and $\boldsymbol{P}$ respectively), and (3) invert the nonlinear relations in order to find the activation (this last step requires to know the kinematic trajectory, and thus $(\boldsymbol{l}, \dot{\boldsymbol{l}})$). It is important to note that in the activation model not only the total force (given by the sum of active and passive forces) must be positive, but the corresponding activation has to be in the range $[\mathbf{0}, \mathbf{1}]$. A valid torque-to-activation relation has to consider both these constraints.

Another important factor to consider is the influence of the model nonlinearities on the representation of the input as linear combination of synergies. In the force model, for the superposition principle, a force $\boldsymbol{F}_i = \boldsymbol{\Gamma} \boldsymbol{b}_i$ leads to a torque $\boldsymbol{\tau}_i$ calculated as follows:

$$\boldsymbol{\tau}_i = \boldsymbol{L}\boldsymbol{F}_i = \boldsymbol{L}\boldsymbol{\Gamma}\boldsymbol{b}_i = \boldsymbol{L}\sum_{j=1}^{n_\gamma} \boldsymbol{\gamma}_j b_{ij} = \boldsymbol{L}\sum_{j=1}^{n_\gamma} (\boldsymbol{L}^+ \boldsymbol{\tau}_j + \boldsymbol{N}\boldsymbol{w}_j) b_{ij} = \sum_{j=1}^{n_\gamma} \boldsymbol{\tau}_j b_{ij} \tag{5.2}$$

where $\boldsymbol{\gamma}_j$ are the force synergies, $\boldsymbol{b}_i = [\boldsymbol{b}_{i1} \dots \boldsymbol{b}_{in_\gamma}]$ are the mixing coefficients, $\boldsymbol{L}$ and $\boldsymbol{N}$ the lever arm and its nullspace matrix respectively, $\boldsymbol{w}_j$ is the nullshift of the synergy $\boldsymbol{\gamma}_j$, and $\boldsymbol{\tau}_j$ is its corresponding torque. The final torque $\boldsymbol{\tau}_i$ is therefore a linear combination of the torques $\boldsymbol{\tau}_j$ associated to the synergies $\boldsymbol{\gamma}_j$. The activation model, behaves very differently. While

in the force model the torque-to-force relation characterized by the pseudoinverse of the lever arm was state-independent (because we considered a state-independent lever arm), and therefore it was equal for each synergy and task, this time the torque-to-activation relations $\boldsymbol{g}$ depend on the particular kinematic trajectory to be translated into activation (see dependence on $(\boldsymbol{l}, \dot{\boldsymbol{l}})$ in Eq. (5.1)). Say that the activation $\boldsymbol{M}_i = \boldsymbol{\Psi}\boldsymbol{c}_i$ is a linear combination of the activation synergies $\{\boldsymbol{\psi}_j\}_{j=1}^{n_\psi}$ through the mixing coefficients vector $\boldsymbol{c}_i$. Say, then, that $\boldsymbol{g}_i$ is the state-dependent torque-to-activation relation associated to the kinematic trajectory corresponding to $\boldsymbol{M}_i$. Similarly, $\boldsymbol{g}_j$ is the torque-to-activation relation of the synergy $\boldsymbol{\psi}_j$. The total torque $\boldsymbol{\tau}_i$ generated by $\boldsymbol{M}_i$ is given by:

$$\boldsymbol{\tau}_i = \boldsymbol{g}_i^{-1}(\boldsymbol{M}_i) = \boldsymbol{g}_i^{-1}(\boldsymbol{\Psi}\boldsymbol{c}_i) = \boldsymbol{g}_i^{-1}(\sum_{j=1}^{n_\psi} \boldsymbol{\psi}_j c_{ij}) = \boldsymbol{g}_i^{-1}(\sum_{j=1}^{n_\psi} \boldsymbol{g}_j(\boldsymbol{\tau}_j) c_{ij}) \quad (5.3)$$

This expression cannot be further simplified because $\boldsymbol{g}_j$ is nonlinear, and because for the above considerations, $\boldsymbol{g}_i^{-1}$ is not the inverse relation of any $\boldsymbol{g}_j$. The superposition principle does not hold and a linear combination of activation synergies does not lead to a corresponding linear combination of torque actuations.

In the next section we derive the torque-to-activation relationship. We find that its form is very similar to that of the force model but its elements, instead of being constant, depend on the kinematic state.

5.2 Nullspace

In this section we define the torque-to-activation relation and suggest a mathematical formulation which, similarly to the force model, allows us to use a nullshift to modulate an activation without changing its corresponding torque. The nullshift can then be used to tackle redundancy and satisfy biological constraints. First, however, we show another possible inversion that deserves attention, and that wasn't used for reasons we will discuss.

We said in the previous section that the activation-to-force relation for a given trajectory is one-to-one. As a result an activation $\boldsymbol{M}_i$ associated to a given torque $\boldsymbol{\tau}_i$ can be obtained by first calculating the force $\boldsymbol{F}_i$ associated to $\boldsymbol{\tau}_i$ (i.e. finding $\boldsymbol{w}_i$ as explained in section 4.2), and then translating $\boldsymbol{F}_i$ to $\boldsymbol{M}_i$. The latter can be done by simply matching Eqs. (2.45) and (4.5) with the constraint that the total force is bigger than the passive element

contribution[2]:

$$\begin{aligned} \boldsymbol{C}_i\boldsymbol{M}_i + \boldsymbol{P}_i &= \boldsymbol{L}^+\boldsymbol{\tau}_i + \boldsymbol{N}\boldsymbol{w}_i \Rightarrow \\ \Rightarrow \boldsymbol{M}_i &= \boldsymbol{C}_i^{-1}(\boldsymbol{L}^+\boldsymbol{\tau}_i + \boldsymbol{N}\boldsymbol{w}_i - \boldsymbol{P}_i) \\ \text{with: } & \boldsymbol{L}^+\boldsymbol{\tau}_i + \boldsymbol{N}\boldsymbol{w}_i \geq \boldsymbol{P}_i \end{aligned} \tag{5.4}$$

This equation leads to a valid activation by definition (we will show later that $\boldsymbol{C}_i$ is invertible in most cases), but it requires to calculate the nullshifts at the force level without considering the nonlinearities coming from $\boldsymbol{C}_i$ and $\boldsymbol{P}_i$, which instead are important when it comes to approximate the desired activations by means of linear combinations of activation synergies.

Another possibility is to find a direct relation between torque and activation without computing the force as an intermediate step. To do so, we define the *active torque* $\boldsymbol{\sigma}_i$ as the torque generated by the active force by means of the CE, without considering the passive muscle force $\boldsymbol{P}_i$. The expression of the active torque $\boldsymbol{\sigma}_i$ as a function of the activation $\boldsymbol{M}_i$ can be derived from Eq. (2.48):

$$\boldsymbol{\sigma}_i = \boldsymbol{\tau}_i - \boldsymbol{L}\boldsymbol{P}_i = \boldsymbol{L}\boldsymbol{C}_i\boldsymbol{M}_i = \boldsymbol{R}_i\boldsymbol{M}_i \tag{5.5}$$

where $\boldsymbol{\tau}_i$ is the total torque (active and passive) and $\boldsymbol{R}_i = \boldsymbol{L}\boldsymbol{C}_i \in \mathbb{R}^{2\times 6}$ is the state-dependent activation-to-torque relation. The subscript i indicates that the relation is different for each task (i.e. state-dependent). From the definition of $\boldsymbol{R}_i$ it follows that its nullspace, defined as $\boldsymbol{Q}_i = \ker(\boldsymbol{R}_i) \in \mathbb{R}^{6\times 4}$, is non-empty and state-dependent. All possible activations $\boldsymbol{M}_i$ leading to an active torque $\boldsymbol{\sigma}_i$ are therefore given by:

$$\boxed{\boldsymbol{M}_i = \boldsymbol{R}_i^+\boldsymbol{\sigma}_i + \boldsymbol{Q}_i\boldsymbol{z}_i} \tag{5.6}$$

where $\boldsymbol{R}_i^+ \in \mathbb{R}^{6\times 2}$ is the Moore-Penrose pseudoinverse of $\boldsymbol{R}_i$ and $\boldsymbol{z}_i \in \mathbb{R}^4$ is the activation *nullshift* which can be freely changed without affecting the torque $\boldsymbol{\sigma}_i$. Since the total torque $\boldsymbol{\tau}_i$ differs from $\boldsymbol{\sigma}_i$ by a quantity that depends only on the kinematic trajectory, which is known a priori, we can state that $\boldsymbol{z}_i$ can be changed without affecting the total torque $\boldsymbol{\tau}_i$. All possible activations $\boldsymbol{M}_i$ leading to a particular torque $\boldsymbol{\tau}_i$ can be found by varying $\boldsymbol{z}_i$.

As for the force model, Eq. (5.6) can be generalized to the synergies. Given the active torque $\boldsymbol{\sigma}_j$ produced by a synergy $\boldsymbol{\psi}_j$, the synergy can be written as:

$$\boldsymbol{\psi}_j = \boldsymbol{R}_j^+\boldsymbol{\sigma}_j + \boldsymbol{Q}_j\boldsymbol{z}_j \tag{5.7}$$

[2]From Eq. 2.45 and from the fact that $\boldsymbol{C}_i$ is positive by definition it follows that $\boldsymbol{F}_i < \boldsymbol{P}_i \Rightarrow \boldsymbol{C}_i\boldsymbol{M}_i + \boldsymbol{P}_i < \boldsymbol{P}_i \Rightarrow \boldsymbol{C}_i\boldsymbol{M}_i < \boldsymbol{0} \Rightarrow \boldsymbol{M}_i < \boldsymbol{0}$ which is not a valid activation. Therefore it must be $\boldsymbol{F}_i \geq \boldsymbol{P}_i$.

for some nullshift $\boldsymbol{z}_j$. Generalizing to all synergies we can write:

$$\boxed{\boldsymbol{\Psi} = \boldsymbol{R}^{+}\boldsymbol{S} + \boldsymbol{Q}\boldsymbol{Z}} \tag{5.8}$$

where $\boldsymbol{\Psi} = [\boldsymbol{\psi}_1 \dots \boldsymbol{\psi}_{n_\psi}] \in \mathbb{R}^{6\times n_\psi}$ is the formal matrix containing the synergies arranged in columns, $\boldsymbol{R}^{+} = [\boldsymbol{R}_1^{+} \dots \boldsymbol{R}_{n_\psi}^{+}] \in \mathbb{R}^{6\times 2n_\psi}$ is the matrix containing the pseudoinverses of $\boldsymbol{R}_j$ arranged in columns, $\boldsymbol{S} = diag(\boldsymbol{\sigma}_j) \in \mathbb{R}^{2n_\psi \times n_\psi}$ is a block diagonal formal matrix with the synergy active torques on the diagonal, $\boldsymbol{Q} = [\boldsymbol{Q}_1 \dots \boldsymbol{Q}_{n_\psi}] \in \mathbb{R}^{6\times 4n_\psi}$ is the matrix with the nullspaces of $\boldsymbol{R}_j$ arranged in columns and finally $\boldsymbol{Z} = diag(\boldsymbol{z}_j) \in \mathbb{R}^{4n_\psi \times n_\psi}$ represents the nullshifts of the individual synergies, arranged on the block diagonal.

Observation It is worth nothing that Eq. (5.6) is very similar to the torque-to-force relation in Eq. (4.5). $\boldsymbol{R}_i$ and $\boldsymbol{Q}_i$ have similar meaning to $\boldsymbol{L}$ and $\boldsymbol{N}$, but are state-dependent. Indeed, they are obtained as the modulation of their force counterpart through the relation $\boldsymbol{C}_i$. The activation model is, therefore, conceptually similar to a model in which the lever arm matrix is state-dependent.

Observation The rank of any $\boldsymbol{R}_i$, and therefore the dimensionality of its nullspace $\boldsymbol{Q}_i$, does not depend on the kinematic state. From the definition of $\boldsymbol{C}_i$ where each diagonal element is $f_l(l)f_v(\dot{l})F_{max}$ and it refers to a particular muscle, $\boldsymbol{R}_i$ is just the product of each column of $\boldsymbol{L}$ by a different positive number. Therefore, if the diagonal elements of $\boldsymbol{C}_i$ are always different from zero, the linear independence of the columns of $\boldsymbol{L}$ and the dimension of the nullspace are preserved. We can observe that:

- $f_l(l)$, the force-length relation, is a gaussian and therefore is different from 0;
- $f_v(\dot{l})$, the force-velocity relation, is equal to 0 iif $\dot{l} = V_{max}$;
- F_{max} is a constant.

The dimension of the nullspace therefore may change only if some muscles are at their maximal contraction velocity. However, in any agonist-antagonist pair, muscles cannot reach their maximum velocities simultaneously if joint velocity is different than zero, because one muscle extends and the other flexes. In isometric contraction instead velocity is zero and the force-velocity relation $f_v(\dot{l})$ is equal to 1. Therefore, for any kinematic trajectory, at least three elements of $\boldsymbol{C}_i$ will be different from zero, and at least three column of $\boldsymbol{R}_i$ (one corresponding to the elbow, one to the shoulder and

one to a biarticular muscle) will be linearly independent, thus preserving the rank 2 of $\boldsymbol{L}$. If the rank would not be preserved, it would mean that for the interested kinematic trajectories we would have additional redundancies. For the same observations above, the $\boldsymbol{C}_i$ is invertible when no muscle is at its maximum contraction velocity. In the following we will assume to be always in this situation.

Like in the force model, suitable torque-to-activation relations have to be found across all possible nullshifts $\boldsymbol{z}_i$ and $\boldsymbol{Z}$. We started by analyzing the dimensionality of the inputs to be approximated, making use of SVD (Singular Value Decomposition). Starting from the same $n = 8$ torque profiles already used in the force model, we generated $m = 100$ linear combinations, obtaining a dataset $\{\boldsymbol{\tau}_i\}_{i=1}^{m}$ of torque actuations with dimensionality n. Then, we calculated $h = 100$ corresponding activation datasets. We have done this in three different ways:

1. using a *fixed* nullshift $\boldsymbol{z}_j$, equal for each activation, but different across the datasets. The activation $\boldsymbol{M}_{ij}$ corresponding to the torque $\boldsymbol{\tau}_i$ in the dataset j is:

$$\boldsymbol{M}_{ij} = \boldsymbol{R}_i^{+}\boldsymbol{\sigma}_i + \boldsymbol{Q}_i\boldsymbol{z}_j \quad \text{for } i = 1 \dots m,\ j = 1 \dots h \tag{5.9}$$

2. using a *variable* nullshift $\boldsymbol{z}_{ij}$, different across both the activations and the datasets. The force $\boldsymbol{M}_{ij}$ corresponding to the torque $\boldsymbol{\tau}_i$ in the dataset j is:

$$\boldsymbol{M}_{ij} = \boldsymbol{R}_i^{+}\boldsymbol{\sigma}_i + \boldsymbol{Q}_i\boldsymbol{z}_{ij} \quad \text{for } i = 1 \dots m,\ j = 1 \dots h \tag{5.10}$$

3. without nullshift ($\boldsymbol{z}_{ij} = \boldsymbol{0}\ \forall i, j$). The force $\boldsymbol{M}_i$ corresponding to the torque $\boldsymbol{\tau}_i$ is:

$$\boldsymbol{M}_i = \boldsymbol{R}_i^{+}\boldsymbol{\sigma}_i \quad \text{for } i = 1 \dots m \tag{5.11}$$

The idea behind the last case, without using any nullshift, is to investigate the intrinsic dimensionality of the torque-to-activation relation. If translating torques to activations increases the dimensionality, the number of required synergies in the activation model may be higher than in the torque model. Note that we didn't do the same test in the force model because the torque-to-force relation (see Eq. (4.5)) is linear and therefore, if $\boldsymbol{w}_{ij} = \boldsymbol{0}$, the dimensionality is preserved.

As in the force model, in each of the above cases, the singular values overlap almost perfectly across datasets. Therefore, in Figure 5.2 we show

only the mean normalized singular values against the number of singular values. The blue line represents the original torque dataset, the dimensionality of which results $n = 8$ as expected, the continuous magenta line represents the activation datasets calculated with a fixed nullshift, while the green line represents those calculated with a variable nullshift. Finally the dashed ma-

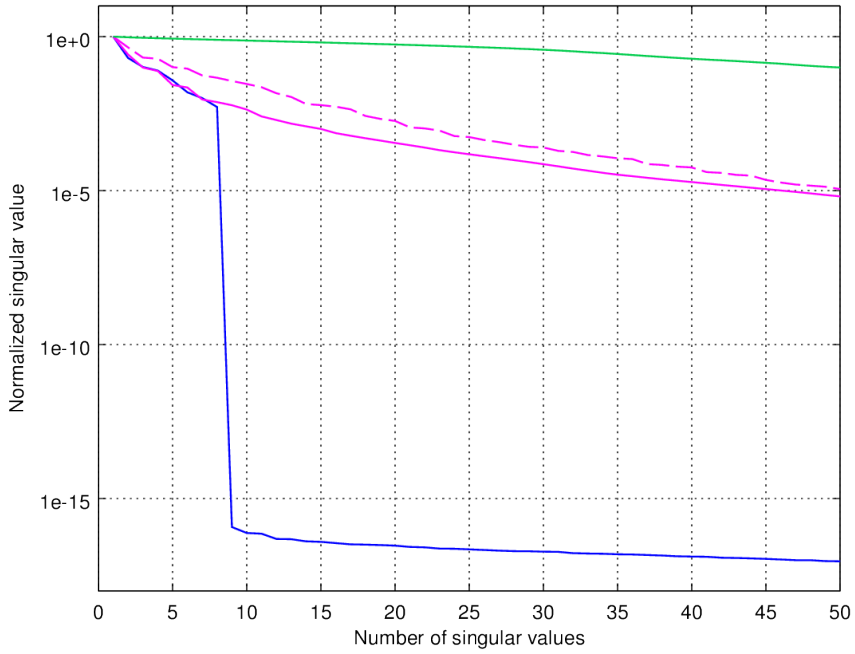


Figure 5.2: Dimensionality comparison between torque dataset (blue line), non-nullshifted activation dataset (dashed magenta line), fixed nullshift activation datasets (continuous magenta line) and variable nullshift activation datasets (green line). Singular values in the two latter cases are the mean of the singular values across all the datasets. The singular values are normalized between 0 and 1.

genta line represents the dimensionality of the activation dataset when no nullshift is used. This dataset has a much higher dimensionality than the torques dataset, and comparable to that of the fixed nullshift datasets. This means that the matrices $\boldsymbol{C}_i$ and $\boldsymbol{P}_i$ themselves lead to an increase of dimensionality. The dimensionality of the variable nullshift datasets, as expected, is higher.

For completeness we show the same comparison in the case where the torque dataset is not generated by linearly combining n generators, but composed by solutions to 100 reaching tasks randomly distributed in the end effector space. Figure 5.3 shows the results, which as expected are very

similar to those obtained in the previous case.

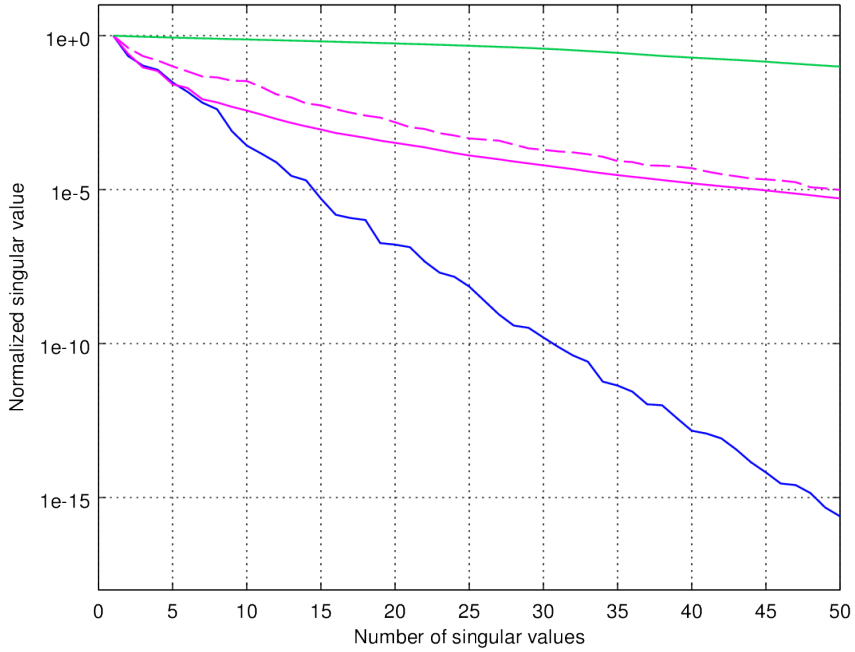


Figure 5.3: Dimensionality comparison between a torque dataset composed of 100 reaching task actuations (blue line), non-nullshifted activation dataset (dashed magenta line), fixed nullshift activation datasets (continuous magenta line) and variable nullshift activation datasets (green line). Singular values in the two latter cases are the mean of the singular values across all the datasets. The singular values are normalized between 0 and 1.

To conclude this analysis we stress again that the high dimensionality of the activation model is provided by muscle nonlinearities through the matrices $\boldsymbol{C}_i$ and $\boldsymbol{P}_i$. This means that more generators are needed to explain the variability in the data, and therefore that we may need more synergies in order to obtain a certain performance, compared to the force and torque models.

5.3 Problem formulation

In this section we formalize the general problem of finding the nullshifts $\boldsymbol{z}_i$ and $\boldsymbol{Z}$ in order to obtain a good approximation of the set of activations solving the desired tasks. We follow the same reasoning we used in the force model. We want to approximate a desired activation $\boldsymbol{M}_i$ with a linear

combination of synergies $\tilde{\boldsymbol{M}}_i = \boldsymbol{\Psi}\boldsymbol{c}_i$ which minimizes the error:

$$\boldsymbol{\varepsilon}_M = \tilde{\boldsymbol{M}}_i - \boldsymbol{M}_i = \boldsymbol{\Psi}\boldsymbol{c}_i - \boldsymbol{M}_i = (\boldsymbol{R}^+\boldsymbol{S} + \boldsymbol{Q}\boldsymbol{Z})\boldsymbol{c}_i - \boldsymbol{R}_i^+\boldsymbol{\sigma}_i - \boldsymbol{Q}_i\boldsymbol{z}_i \quad (5.12)$$

where $\boldsymbol{c}_i = [c_{i1} \dots c_{in_\psi}] \in \mathbb{R}^{n_\psi}$ is a vector of mixing coefficients. This problem can be formulated as:

$$\begin{aligned}[\boldsymbol{c}_i, \boldsymbol{z}_i, \boldsymbol{Z}] &= \underset{\boldsymbol{c}_i,\boldsymbol{z}_i,\boldsymbol{Z}}{\operatorname{argmin}} \|\boldsymbol{\Psi}\boldsymbol{c}_i - \boldsymbol{M}_i\| \\ &= \underset{\boldsymbol{c}_i,\boldsymbol{z}_i,\boldsymbol{Z}}{\operatorname{argmin}} \|(\boldsymbol{R}^+\boldsymbol{S} + \boldsymbol{Q}\boldsymbol{Z})\boldsymbol{c}_i - \boldsymbol{R}_i^+\boldsymbol{\sigma}_i - \boldsymbol{Q}_i\boldsymbol{z}_i\| \\ &\quad \text{s.t.} \quad (\boldsymbol{R}^+\boldsymbol{S} + \boldsymbol{Q}\boldsymbol{Z})\boldsymbol{c}_i \geq \boldsymbol{0}\end{aligned} \quad (5.13)$$

where the norm $\|\cdot\|$ applied to a function $\boldsymbol{f}(t)$ must be interpreted as follows:

$$\|\boldsymbol{f}(t)\| = \sqrt{\int_{t\in T} |\boldsymbol{f}(t)|^2 \mathrm{d}t}$$

As discussed in the force model (section 4.3) one of the main assumption of the synergy hypothesis is that synergies are invariant across tasks, which means that the nullshift $\boldsymbol{Z}$ does not change. From Eq. (5.12) and given that $\boldsymbol{Q}_i^+\boldsymbol{R}_i^+ = \boldsymbol{0}$, all possible solutions that minimize $\boldsymbol{\varepsilon}_M$ can be expressed as follows:

$$\boldsymbol{z}_i = \boldsymbol{Q}_i^+(\boldsymbol{R}^+\boldsymbol{S}\boldsymbol{c}_i + \boldsymbol{Q}\boldsymbol{Z}\boldsymbol{c}_i - \boldsymbol{R}_i^+\boldsymbol{\sigma}_i) = \boldsymbol{Q}_i^+(\boldsymbol{R}^+\boldsymbol{S} + \boldsymbol{Q}\boldsymbol{Z})\boldsymbol{c}_i \quad (5.14)$$

Substituting this value of $\boldsymbol{z}_i$ in Eq. (5.13) and assuming a given $\boldsymbol{Z}$, the final expression of the general problem is:

$$\begin{aligned}\boldsymbol{c}_i &= \underset{\boldsymbol{c}_i}{\operatorname{argmin}} \|(\boldsymbol{R}^+\boldsymbol{S} + \boldsymbol{Q}\boldsymbol{Z})\boldsymbol{c}_i - \boldsymbol{R}_i^+\boldsymbol{\sigma}_i - \boldsymbol{Q}_i\boldsymbol{Q}_i^+(\boldsymbol{R}^+\boldsymbol{S} + \boldsymbol{Q}\boldsymbol{Z})\boldsymbol{c}_i\| \\ &= \underset{\boldsymbol{c}_i}{\operatorname{argmin}} \|(\boldsymbol{I} - \boldsymbol{Q}_i\boldsymbol{Q}_i^+)(\boldsymbol{R}^+\boldsymbol{S} + \boldsymbol{Q}\boldsymbol{Z})\boldsymbol{c}_i - \boldsymbol{R}_i^+\boldsymbol{\sigma}_i\| \\ &\quad \text{s.t.} \quad (\boldsymbol{R}^+\boldsymbol{S} + \boldsymbol{Q}\boldsymbol{Z})\boldsymbol{c}_i \geq \boldsymbol{0}\end{aligned} \quad (5.15)$$

Differently from the force model, both the constraint and the cost function depend on the particular choice of $\boldsymbol{Z}$ for the activation synergies. In the following section we show a possible strategy to choose such a $\boldsymbol{Z}$.

5.4 Identifying synergies

In the previous section we proposed a method to calculate the optimal mixing coefficients $\boldsymbol{c}_i$ and the optimal nullshift $\boldsymbol{z}_i$ in order to minimize the projection error. In this section we introduce some possible strategies to define the synergy nullshift $\boldsymbol{Z}$.

As for the force model, a first choice regards whether synergies are valid actuations (i.e. satisfy the positivity constraints), or not. In section 4.4.1 we argued, supported by test results, that in general it is not possible to approximate positive actuations with unconstrained synergies, except if they have a very particular form. For the same reasons we assume here that the nullshift $\boldsymbol{Z}$ must lead to positive synergies. We propose two models for $\boldsymbol{Z}$:

A The same nullshift is applied to all synergies such that they are positive.

B A variable nullshift is applied to each synergy such that they are positive.

In the following, initially we formalize the two models, and then we show the results for both of them.

5.4.1 Model A: Same nullshift with positivity constraints

The nullshift of the synergies is defined as:

$$\boldsymbol{Z} = diag(\boldsymbol{z}^* \dots \boldsymbol{z}^*) \quad \text{such that} \quad \boldsymbol{\Psi} \geq \boldsymbol{0} \tag{5.16}$$

The nullshift $\boldsymbol{z}^*$ was calculated in the same way as the $\boldsymbol{w}^*$ in the force model, using the following linear optimization problem:

$$\begin{aligned} \boldsymbol{z}^*(t) = \underset{\boldsymbol{z}^*(t)}{\operatorname{argmin}} & \sum_t \boldsymbol{Q}_i \boldsymbol{z}^*(t) \\ \text{s.t.} \quad & \boldsymbol{R}_i^+ \boldsymbol{\sigma}_i(t) + \boldsymbol{Q}_i \boldsymbol{z}^*(t) \geq \boldsymbol{0} \ \text{ for } \ i = 1..200 \end{aligned} \tag{5.17}$$

where the constraints are set on the activation corresponding to the active torque $\boldsymbol{\sigma}_i$ of 200 representative reaching tasks uniformly distributed in the end effector space.

5.4.2 Model B: Variable nullshifts with positivity constraints

In this model the synergy nullshift is defined as:

$$\boldsymbol{Z} = diag(\boldsymbol{z}_1 \dots \boldsymbol{z}_{n_\psi}) \quad \text{such that} \quad \boldsymbol{\Psi} \geq \boldsymbol{0} \tag{5.18}$$

This is the most general formulation of $\boldsymbol{Z}$ and gives the possibility to shape each synergy as required. For example, one may choose to set the $\boldsymbol{z}_j$ in order to increase orthogonality between the synergies, or to decrease the activation magnitude, or something else. However, depending on the particular criteria and on the number of synergies, it can be extremely difficult and

computationally intensive to find an adequate $\boldsymbol{Z}$, furthermore it would require to find all its components at ones. This is in contrast with the method we propose to synthesize synergies, the reduction (see section 2.3), which instead is an iterative process.

Here we propose a possible approach to find each $\boldsymbol{z}_j$ iteratively in a way compatible with the reduction method. The idea is based on the fact that active forces are generated as the product between muscle activations and the nonlinear relation $\boldsymbol{C}$. The proposed approach tries to encapsulate the information about $\boldsymbol{C}$ into the nullshifts $\boldsymbol{z}_j$, and to produce actuations compatible with those of the force model obtained with fixed nullshift $\boldsymbol{w}^*$. From the general torque-to-force relation (4.5) we can define the active force $\boldsymbol{F}_j$ leading to the active torque $\boldsymbol{\sigma}_j$ as:

$$\boldsymbol{F}_j = \boldsymbol{L}^+\boldsymbol{\sigma}_j + \boldsymbol{N}\boldsymbol{w}^* \tag{5.19}$$

for some $\boldsymbol{w}^*$. Note that this $\boldsymbol{w}^*$ may be different than that found in the force model as here, we have to deal with the passive properties of the muscles, thus we have to calculate such a nullshift using the active torque $\boldsymbol{\sigma}_j$ only, and not the total torque $\boldsymbol{\tau}_j$. The relation between the active force $\boldsymbol{F}_j$ and the corresponding activation $\boldsymbol{M}_j$ is (see Eq. (2.45)):

$$\boldsymbol{F}_j = \boldsymbol{C}_j\boldsymbol{M}_j \tag{5.20}$$

Matching these two equations, and using the general torque-to-activation relation in Eq. (5.6) we obtain the expression of the activation nullshift $\boldsymbol{z}_j$ as a function of force nullshift $\boldsymbol{w}^*$ as follows:

$$\begin{aligned}
\boldsymbol{C}_j\boldsymbol{M}_j &= \boldsymbol{L}^+\boldsymbol{\sigma}_j + \boldsymbol{N}\boldsymbol{w}^* \Rightarrow \\
\boldsymbol{C}_j(\boldsymbol{R}_j^+\boldsymbol{\sigma}_j + \boldsymbol{Q}_j\boldsymbol{z}_j) &= \boldsymbol{L}^+\boldsymbol{\sigma}_j + \boldsymbol{N}\boldsymbol{w}^* \Rightarrow \\
\boldsymbol{C}_j\boldsymbol{Q}_j\boldsymbol{z}_j &= (\boldsymbol{L}^+ - \boldsymbol{C}_j\boldsymbol{R}_j^+)\boldsymbol{\sigma}_j + \boldsymbol{N}\boldsymbol{w}^* \Rightarrow \\
\boldsymbol{z}_j &= \boldsymbol{Q}_j^+\boldsymbol{C}_j^{-1}((\boldsymbol{L}^+ - \boldsymbol{C}_j\boldsymbol{R}_j^+)\boldsymbol{\sigma}_j + \boldsymbol{N}\boldsymbol{w}^*) \Rightarrow \\
\boldsymbol{z}_j &= \boldsymbol{Q}_j^+\boldsymbol{C}_j^{-1}(\boldsymbol{L}^+\boldsymbol{\sigma}_j + \boldsymbol{N}\boldsymbol{w}^*)
\end{aligned} \tag{5.21}$$

where the last equality results from the fact that $\boldsymbol{C}_j$ is diagonal and therefore invertible, and $\boldsymbol{Q}_j^+\boldsymbol{R}_j^+ = \boldsymbol{0}$.

The value of $\boldsymbol{w}^*$ has been calculated across 200 representative tasks by running the following optimization:

$$\begin{aligned}
\boldsymbol{w}^*(t) = \underset{\boldsymbol{w}(t)}{\operatorname{argmin}} & \sum_t \boldsymbol{N}\boldsymbol{w}(t) \\
\text{s.t.} \quad & \boldsymbol{L}^+\boldsymbol{\sigma}_i(t) + \boldsymbol{N}\boldsymbol{w}(t) \geq \boldsymbol{0} \;\text{ for }\; i = 1..200
\end{aligned} \tag{5.22}$$

Note that this linear problem is equivalent to (4.24), expect that in this case the force which must be positive is not the total force, but only the active force (leading to the active torque $\boldsymbol{\sigma}_i$, which does not include the contribute from the parallel element of the muscles). A positive active force leads, by definition, to a positive activation, therefore if the force calculated with $\boldsymbol{w}^*$ is positive, then the activation calculated using $\boldsymbol{z}_j$ will also be positive.

We used the force nullshift $\boldsymbol{w}^*$ found as described above to calculate the nullshifts $\boldsymbol{z}_j$ of the synergies. Note however that while $\boldsymbol{w}^*$ is fixed, the nullshift $\boldsymbol{z}_j$ varies across synergies, as it depends on $\boldsymbol{\sigma}_i$ (i.e. active torque of synergy j). Furthermore it is modulated by the state-dependent relations $\boldsymbol{C}_j$ and $\boldsymbol{Q}_j$. The active torque $\boldsymbol{\sigma}_i$ of the synergies are calculated iteratively by means of the reduction, as explained in the next section.

5.5 Reduction

We ran the reduction process for both the models A (fixed nullshift) and B (variable nullshift) presented above.

In model A, at iteration j, the new synergy was calculated from the active torque $\boldsymbol{\sigma}_j$ as follows:

$$\boldsymbol{\psi}_j = \boldsymbol{R}_j^+ \boldsymbol{\sigma}_j + \boldsymbol{Q}_j \boldsymbol{z}^* \tag{5.23}$$

If the new synergy $\boldsymbol{\psi}_j$ was not positive, we improved the value of the nullshift $\boldsymbol{z}^*$ with the same strategy followed in the force model:

$$\begin{gathered} \boldsymbol{z}^* = \boldsymbol{z}_j^* = \underset{\boldsymbol{z}_j}{\operatorname{argmin}} \|\boldsymbol{z}_j - \boldsymbol{z}_{j-1}^*\| \\ \text{s.t.} \quad \boldsymbol{\Psi} \geq \boldsymbol{0} \end{gathered} \tag{5.24}$$

In model B, at iteration j, the new synergy was calculated from the active torque $\boldsymbol{\sigma}_j$ by using Eq. (5.21), as follows:

$$\boldsymbol{\psi}_j = \boldsymbol{R}_j^+ \boldsymbol{\sigma}_j + \boldsymbol{Q}_j \boldsymbol{z}_j = \boldsymbol{R}_j^+ \boldsymbol{\sigma}_j + \boldsymbol{Q}_j \boldsymbol{Q}_j^+ \boldsymbol{C}_j^{-1} (\boldsymbol{L}^+ \boldsymbol{\sigma}_j + \boldsymbol{N} \boldsymbol{w}^*) \tag{5.25}$$

If the new synergy $\boldsymbol{\psi}_j$ was not positive we improved the value of the active force nullshift $\boldsymbol{w}^*$ with:

$$\begin{gathered} \boldsymbol{w}^* = \boldsymbol{w}_j^* = \underset{\boldsymbol{w}_j}{\operatorname{argmin}} \|\boldsymbol{w}_j - \boldsymbol{w}_{j-1}^*\| \\ \text{s.t.} \quad \boldsymbol{\Psi} \geq \boldsymbol{0} \end{gathered} \tag{5.26}$$

As described in section 2.3, at each reduction step we evaluated the projection errors in approximating the solutions to over 700 reaching tasks

Table 5.1: Prototasks positions in the activation model with synergies nullshifts calculated as in model A. Coordinates are expressed in elbow and shoulder joint angles.

#	q_e [rad]	q_s [rad]
1	2.3679	2.1889
2	0.7669	-0.3771
3	1.2274	2.0619
4	2.3324	-0.1515
5	0.9310	0.0065
6	1.7669	-0.2848
7	2.4551	0.7425
8	1.8115	2.2267
9	1.9760	1.4898
10	2.4069	1.8872
11	1.5979	1.0314
12	2.0476	0.5248

Table 5.2: Prototasks positions in the activation model with synergies nullshifts calculated as in model B. Coordinates are expressed in elbow and shoulder joint angles.

#	q_e [rad]	q_s [rad]
1	2.3679	2.1889
2	0.7669	-0.3771
3	1.2274	2.0619
4	2.3324	-0.1515
5	1.4105	-0.2035
6	1.5006	1.3475
7	2.4150	0.8098
8	1.9577	0.5146
9	2.0103	1.6435
10	1.1814	0.6275
11	1.6550	2.2471
12	0.7660	-0.0104

distributed homogeneously in the end effector space. We didn't evaluate reaching tasks with final positions along the kinematic chain boundaries in order to avoid model instability[3]. The mixing coefficients and the nullshifts $\boldsymbol{z}_i$ of each task were calculated by solving Eqs. (5.15) and (5.14) respectively. The result of the reduction procedure with 12 iterations on models A and B are shown in Figures 5.4 and 5.5 respectively, where panels are arranged in row-major order and each one corresponds to one reduction iteration. Tables 5.1 and 5.2 show the prototask coordinates for the two models.

5.6 Performance

We evaluated the performance of the activation model and compared them with those of the torque and force models. Figures 5.6 and 5.7 show respectively the mean relative projection error and the mean end effector forward dynamics error of the activation model (green and magenta lines) compared

[3]Even though the desired activation leads to a valid trajectory, the activation found by linear combination of synergies may lead to one which instead exceeds the boundaries. In this case muscles exceed their maximum length, causing unpredictable effects (e.g. passive force diverges). Such effects can be avoided for example modifying the dynamics of the kinematic chain and/or of the muscles.

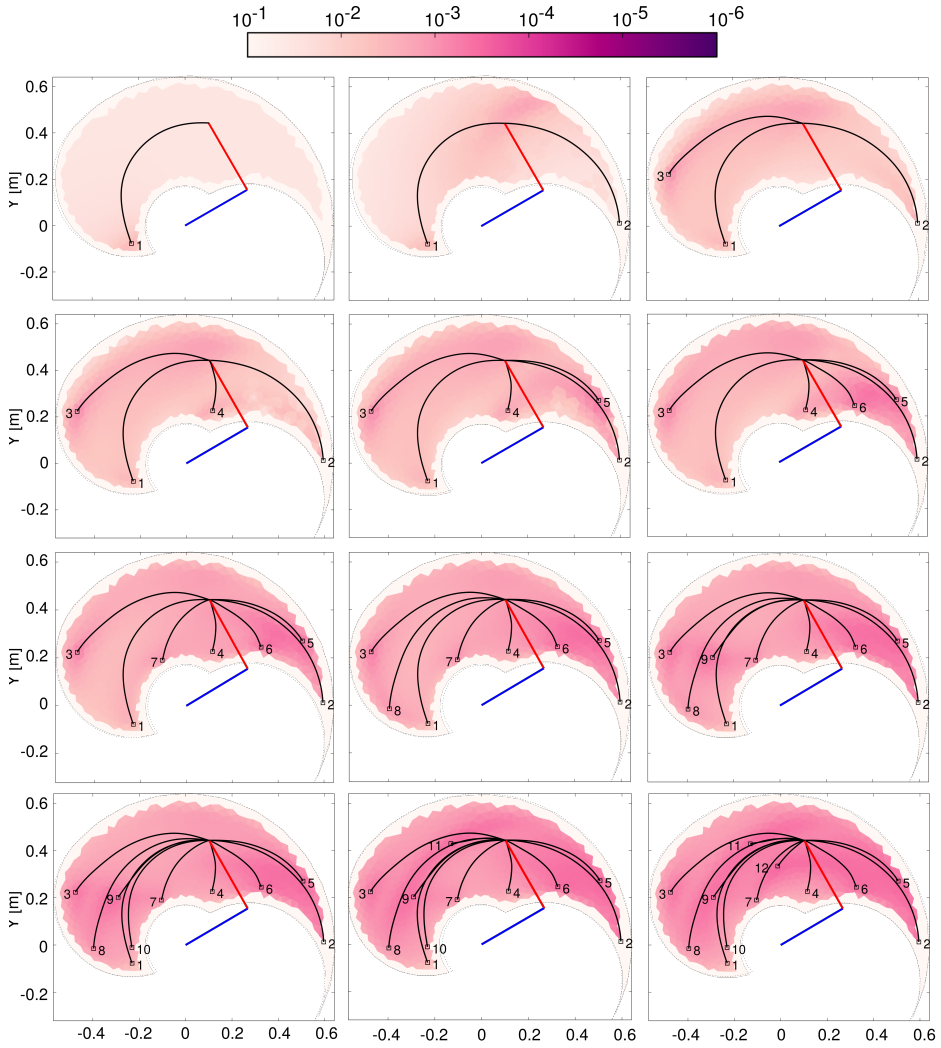


Figure 5.4: Results of the reduction procedure for the kinematic chain actuated with fixed nullshift activation (model A). Each panel corresponds to a number of synergies. Each point of the operational space is colored depending on the performance of the synergies in approximating a reaching task solution with final position in that point. Bright areas correspond to high projection errors while dark areas correspond to small projection errors as depicted in the color bar above. White areas on the boundaries of the operational space were not evaluated to avoid model instability in case of projected activations leading to trajectories exceeding the boundaries.

to the results obtained in the torque model (blue line) and in the force model (red line). The green line corresponds to the activation model with fixed synergies nullshifts (model A), while the magenta line corresponds to the model with variable synergies nullshifts (model B). As expected, the

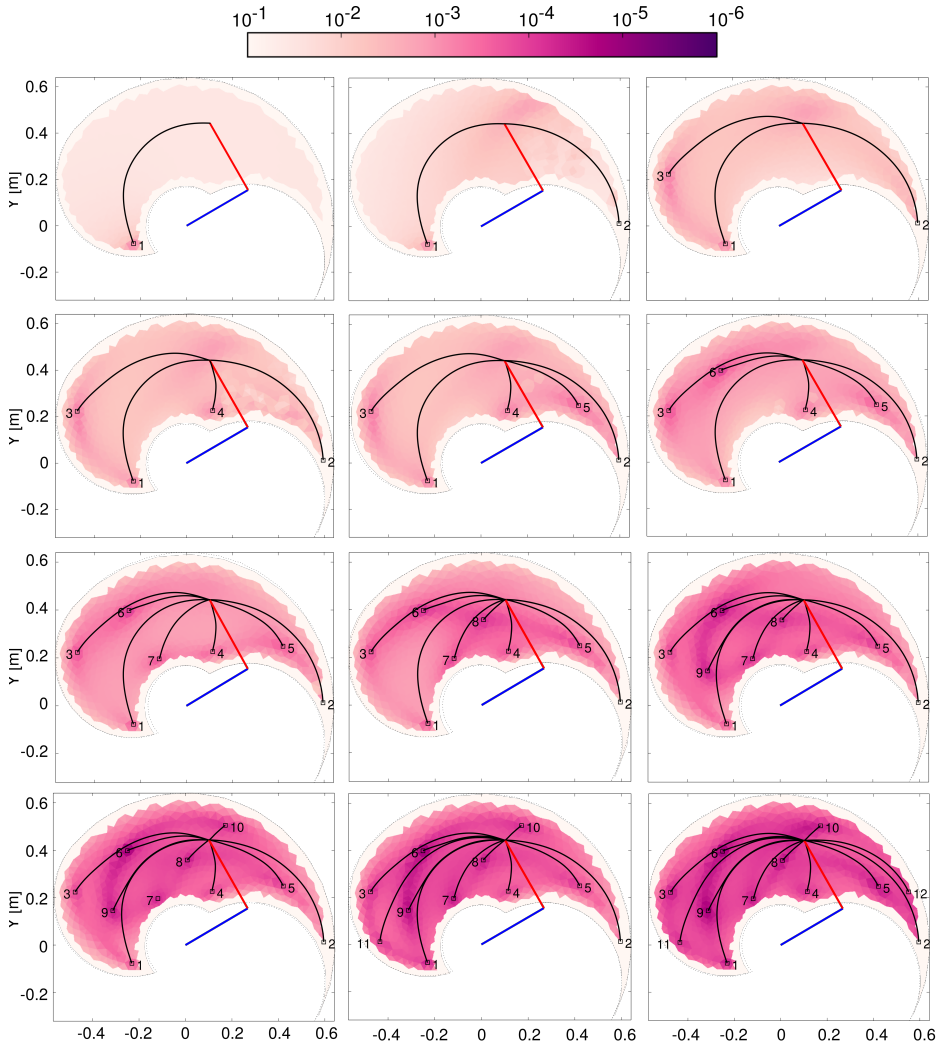


Figure 5.5: Results of the reduction procedure for the kinematic chain actuated with variable nullshift activation (model B). Each panel corresponds to a number of synergies. Each point of the operational space is colored depending on the performance of the synergies in approximating a reaching task solution with final position in that point. Bright areas correspond to high projection errors while dark areas correspond to small projection errors as depicted in the color bar above. White areas on the boundaries of the operational space were not evaluated to avoid model instability in case of projected activations leading to trajectories exceeding the boundaries.

activation model requires more synergies than the torque and force models to obtain the same performance. The reason for this is the explosion of the dimensionality at the activation level, described in section 5.2. There is a non-negligible difference in performance between the activation models

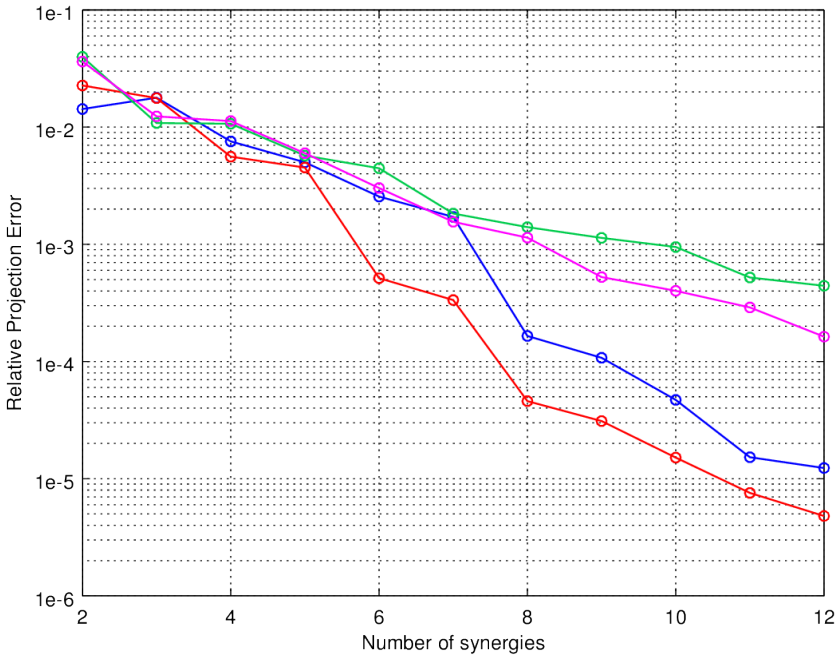


Figure 5.6: Relative projection error vs. number of synergies for the activation model with fixed nullshift (green line) and variable nullshift (magenta line), compared to the torque model (blue line) and the force model (red line).

with fixed and variable nullshifts. The errors of the fixed nullshift model are more than half order of magnitude higher than those of the variable nullshift model. This is in accordance with our consideration that since the nullspace $\boldsymbol{Q}_i$ is state-dependent, a fixed nullshift does not necessarily lead to better results (as in the force model) and that a suitable choice of the synergies nullshifts $\boldsymbol{Z}$ may lead to more satisfactory results.

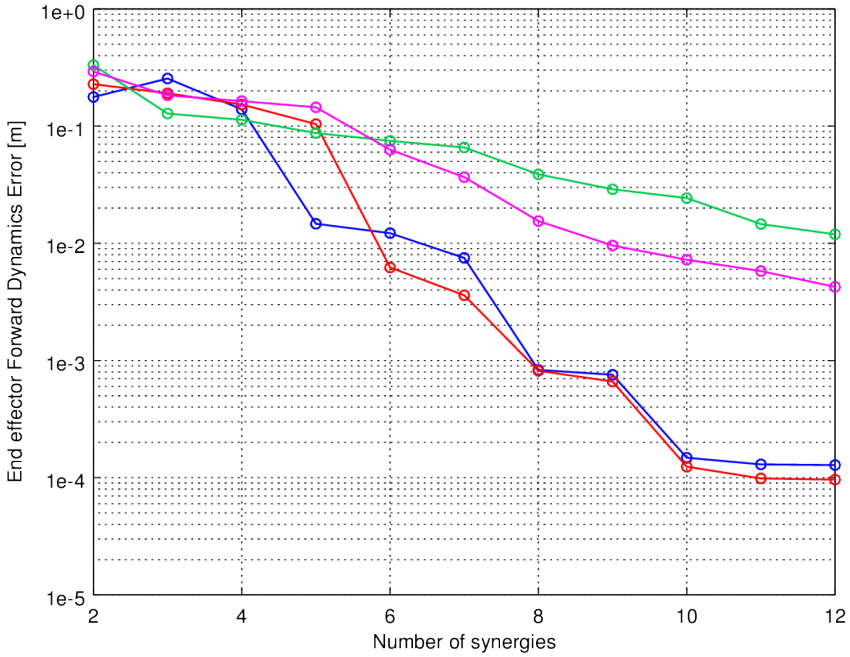


Figure 5.7: End effector forward dynamics error vs. number of synergies for the activation model with fixed nullshift (green line) and variable nullshift (magenta line), compared to the torque model (blue line) and the force model (red line).

Chapter 6

Conclusions

6.1 Thesis Contributions

In this work we investigated the muscle synergy hypothesis following a computational approach. Many researchers provided indirect evidence for the existence of muscle synergies, mostly by analyzing EMG recordings and by extracting components that explain the dataset variability. This approach has provided a lot of insights, but many questions remain open. Among others, one very important question is whether muscle synergies are a valid control strategy from the point of view of the musculoskeletal dynamics. Trying to address these questions, we investigated (1) the extent to which redundancy and muscle nonlinearities influence the muscle synergy hypothesis, and (2) whether a musculoskeletal model that features these properties can be controlled by linear combinations of synergies. In order to do so we followed a computational approach. We defined a model of human arm composed by 2 joints representing shoulder and elbow. We started with a simple model, and systematically added redundancy and nonlinearities: initially we actuated the kinematic chain with torques applied to the joints, leading to a non-redundant model; then we generated the torques my means of 6 forces applied to the links, thus adding redundancy; finally we generated the forces my means of nonlinear muscle models controlled in muscle activation. In each of the mentioned models, we synthesized appropriate time-varying synergies, i.e. signals in the input space of the model (torques, forces and muscle activations). Synergies were synthesized through a procedure based on the DRD (Dynamic Response Decomposition) method, called reduction. We then evaluated how well linear combinations of the synthesized synergies can approximate a set of desired actuations, and let the system perform the corresponding reaching tasks. The reaching tasks were distributed in the

entire end effector space, and were defined as the movement of the joints from an initial position to a final position, with initial and final velocities equal to zero.

In the torque model, we were able to perform reaching tasks by linearly combining 8 synergies. We could approximate the desired actuations with a mean relative approximation error (i.e. relative projection error) of around 10^{-4} (see Figure 5.6, blue line). We obtained a mean error in the end effector position lower than $1mm$ (see Figure 5.7, blue line), meaning that the distance between the desired and the obtained final position was smaller than $1mm$. These results were similar to those obtained in previous research.

In the force model, redundancy requires to choose among an infinite number of forces which lead to the same torque, thus to the same kinematic trajectory. Furthermore, due to biological constraints on the muscles, forces have to satisfy a positivity constraint. Despite this added complexity, we showed that if synergies and desired forces (i.e. those solving the reaching tasks) are calculated in a particular way (i.e. fixed nullshift), the dimensionality of the force dataset to be approximated, is similar to that of the corresponding torque dataset. With the same number of synergies we were able to obtain similar performance to those of the torque model (see Figure 5.7, red line). These results suggest that muscle redundancy does not necessary lead to a increase in the complexity of the control by linear combination of synergies. This means that, for a given number of degrees of freedom, the number of synergies required by the CNS may not depend on the number of muscles to be controlled.

In the activation model the synergies are represented as muscle activation profiles. Their linear combinations represent the input to nonlinear muscle models, which in turn produce the forces applied to the kinematic chain. We showed that this model requires more synergies than the force and torque models, to obtain the same performance (see Figure 5.7, magenta line). Furthermore, performance increases with the number of synergies at a lower rate than in the other models, meaning that the gain in performance obtained when adding a new synergy is smaller in the activation model than in the other two. These results are supported by the fact that the dimensionality of the desired activations (solutions to the reaching tasks) is much higher than the dimensionality of the desired forces and desired torques, as shown in section 5.2. This increase of dimensionality derives from the muscles dynamics. These results suggest that muscle nonlinearity may increase the required number of synergies to obtain a given performance, and may represent a challenge for a synergy based controller. Similar results have been obtained in [56]. During reaching tasks, the authors estimated torque

actuations from joint kinematics. They also extracted synergies by means of EMG recordings and found that the number of generators (i.e. synergies) needed to explain the variability in the EMG dataset is higher than the number of generators in the torque dataset.

Finally, we showed that in all three models we could successfully control the kinematic chain in order to perform reaching tasks. Although we show that biomechanical features influence the number and characteristics of the synergies, our results suggest that muscle synergies may be a feasible strategy for the CNS to simplify the control in musculoskeletal systems. Differently from approaches based on the analysis of EMG signals, which lead to results that are very difficult to validate, we have shown that with a suitable synergy set, it is possible to control a simplified musculoskeletal model of the human arm.

6.2 Discussion and Future Work

Neuroscience

We showed that, depending on the features of the musculoskeletal system, the number of synergies needed in order to obtain a certain performance may be different (i.e. increases in case of nonlinear muscles). This means that the number and characteristics of the synergies may depend on their *level of representation* (kinematics, muscle activation, neural activity etc.). In literature synergies have been investigated at different levels. For example, several authors searched for synergies at the EMG level (similar to our muscle activation) [22, 63, 42, 43], other have found evidence for kinematic synergies [66, 57, 29, 60]. In [60] in particular, the authors have found that during reach and grasp movements, synergies at muscle activation level (i.e. EMG) are also at the origin of kinematic synergies. Our work shows that experimental results may vary considerably depending on the level at which the muscle synergy hypothesis is investigated, therefore we believe that the community should put some effort in investigating what synergies may actually represent. Otherwise the conclusions of experiments may be misleading.

Another important consideration is about the validity of the individual synergies as task solution. It is not clear what synergies really represent and there is no general agreement in neuroscience on this (see for example [18, 48, 12] for different points of view). In our work we ask whether they represent valid inputs, in which case they have to satisfy all biological constraints (e.g. positive muscle activation) and lead to a valid trajectory, or

they are abstract generators defined at some level, which when combined lead to valid motor commands. We showed that, under our assumptions, non-valid synergies do not lead to good approximation of desired actuations, nor they lead to good task performance. In our framework, however, one could think to use the nullshift in order to obtain abstract synergy profiles, which combined lead to valid actuations. We leave this idea for future work. This is nevertheless a key question in neuroscience: if the synergies can be anything (i.e. abstract generators) then we may be able to find them at any level of description, and they only have to satisfy the requirement that they reconstruct some recorded dataset. If we say, instead, that synergies must have particular characteristics (i.e. valid actuations) we reduce the set of possible representations. It would therefore be easier to challenge the synergy hypothesis with specific experiments, trying to falsify it. This analysis is left for future work.

In section 4.6 we showed that both relative projection error and end effector error of the force model are, for most number of synergies, below the respective errors of the torque model (see Figures 4.10 and 4.11). We speculate that this happens because the synergy controller may exploit redundancy in order to better approximate the desired inputs, given that it has more degrees of freedom available. Additional work has to be done to investigate this speculation.

Methodology

The procedure we used to synthesize synergies (reduction) is driven by the projection error. However, during our experiments we could observe that in some cases the projection error does not represent a suitable performance measure. Indeed the projection error measures the ability of the synergies in approximating a desired input. When the musculoskeletal system is redundant, however, part of the input does not lead to any change in the kinematic state (for the force model, this happens if portion of input is in the nullspace of the lever arm matrix, see section 4.2). The result is that we measure, at least in part, something that does not have any effect on the task performance, and the result can be very misleading. These considerations do not apply only to the computational approach presented here, but also to more classical approaches like the analysis of EMG dataset, where researchers measure the ability of the extracted synergies to reconstruct a dataset (i.e. they measure an approximation error). Synergy extraction methods should explicitly take into account task execution variables [3]. A

possible direction for future work is therefore that of trying to drive the reduction procedure using a direct task performance measure (i.e. forward dynamics error).

In this work we considered a simplified model of the human arm. The lever arm was fixed, and muscles were composed by the contractile and parallel elements. We did not include the serial element, which accounts for compliance during contractions, nor a variable lever arm, or friction at the joints and so on. These elements add additional complexity and therefore they might lead to further increase in the number of synergies needed to obtain a certain performance. On the other hand, from where the synergies are assumed to be stored (e.g. brain, spinal cord), to the muscles where they are consumed, the information they carry goes through different complex systems and is possibly influenced by many factors like those mentioned above. It may be, therefore, that some of these factors compensate the complexity introduced by other factors, thus they may not affect, or may even reduce, the number of synergies. Additional work is required to study such interaction effects.

In the activation model we proposed a possible way to calculate the synergy nullshift from the force nullshift. This leads to better performance compared with other methods (i.e. fixed nullshift). Under the assumption that synergies represent task actuations, we ask if there is a set of nullshifts for the synergies which further improves performance. Such a set may exist, but finding it may be non-trivial from both the methodological and the computational point of view. Nevertheless, investigating this issue may be worth it.

Finally, a possible investigation which could be performed in future work consists in trying to see, on a more complete and realistic model, if there is some correlation between the synergies synthesized with the computational approach used here, and the synergies experimentally extracted from EMG datasets. If some correspondence could be found, this would represent a breakthrough for the muscles synergy hypothesis. It would mean that synergies might be solutions to prototypical tasks, and that they embed information on task and body dynamics.

Bibliography

[1] C. Alessandro, J. P. Carbajal, and A. d'Avella. Synthesis and Adaptation of Effective Motor Synergies for the Solution of Reaching Tasks. In T. Ziemke, C. Balkenius, and J. Hallam, editors, *From Animals to Animats 12, 12th International Conference on Simulation of Adaptive Behavior, SAB 2012, Proceedings*, volume 7426 of *Lecture Notes in Computer Science*, pages 33–43. Springer, 2012.

[2] C. Alessandro, J. P. Carbajal, and A. d'Avella. A computational analysis of motor synergies by dynamic response decomposition. *Frontiers in computational neuroscience*, 7:191, 2013.

[3] C. Alessandro, I. Delis, F. Nori, S. Panzeri, and B. Berret. Muscle synergies in neuroscience and robotics: from input-space to task-space perspectives. *Frontiers in Computational Neuroscience*, 7:43, 2013.

[4] K. N. An, F. C. Hui, B. F. Morrey, R. L. Linscheid, and E. Y. Chao. Muscles across the elbow joint: A biomechanical analysis. *Journal of Biomechanics*, 14(10):659–69, 1981.

[5] C. G. Atkeson and J. M. Hollerbach. Kinematic features of unrestrained vertical arm movements. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 5(9):2318–30, 1985.

[6] A. Barhorst and L. Schovanec. A Neuro-Muscular Elasto-Dynamic Model of the Human Arm Part 1: Model Development. *Journal of Bionic Engineering*, 6(2-6):93–107, 2009.

[7] A. Bengoetxea, F. Leurs, T. Hoellinger, A. M. Cebolla, B. Dan, G. Cheron, and J. McIntyre. Physiological modules for generating discrete and rhythmic movements: component analysis of EMG signals. *Frontiers in computational neuroscience*, 8:169, 2014.

[8] D. J. Berger and A. d'Avella. Effective force control by muscle synergies. *Frontiers in computational neuroscience*, 8:46, 2014.

[9] D. J. Berger, R. Gentner, T. Edmunds, D. K. Pai, and A. d'Avella. Differences in adaptation rates after virtual surgeries provide direct evidence for modularity. *The Journal of neuroscience*, 33(30):12384–94, 2013.

[10] M. Berniker, A. Jarc, E. Bizzi, and M. C. Tresch. Simplified and effective motor control based on muscle synergies to exploit musculoskeletal dynamics. *Proceedings of the National Academy of Sciences of the United States of America*, 106(18):7601–7606, 2009.

[11] N. Bernstein. The coordination and regulation of movements, 1967.

[12] E. Bizzi and V. C. Cheung. The neural origin of muscle synergies. *Frontiers in computational neuroscience*, 7:51, 2013.

[13] D. Borzelli, D. J. Berger, D. K. Pai, and A. d'Avella. Effort minimization and synergistic muscle recruitment for three-dimensional force generation. *Frontiers in computational neuroscience*, 7:186, 2013.

[14] G. Cappellini, Y. P. Ivanenko, R. E. Poppele, and F. Lacquaniti. Motor patterns in human walking and running. *Journal of neurophysiology*, 95(6):3426–37, 2006.

[15] J. P. Carbajal. *Harnessing Nonlinearities: Behavior Generation from Natural Dynamics.* PhD thesis, University of Zürich, 2012.

[16] J. W. Chow and W. G. Darling. The maximum shortening velocity of muscle should be scaled with activation. *Journal of Applied Physiology*, 86(3):1025–31, 1999.

[17] S. A. Chvatal and L. H. Ting. Common muscle synergies for balance and walking. *Frontiers in computational neuroscience*, 7:48, 2013.

[18] S. A. Chvatal, G. Torres-Oviedo, S. A. Safavynia, and L. H. Ting. Common muscle synergies for control of center of mass and force in nonstepping and stepping postural behaviors. *Journal of neurophysiology*, 106(2):999–1015, 2011.

[19] M. Coscia, V. C. Cheung, P. Tropea, A. Koenig, V. Monaco, C. Bennis, S. Micera, and P. Bonato. The effect of arm weight support on upper limb muscle synergies during reaching movements. *Journal of neuroengineering and rehabilitation*, 11:22, 2014.

[20] M. D'Andola, B. Cesqui, A. Portone, L. Fernandez, F. Lacquaniti, and A. d'Avella. Spatiotemporal characteristics of muscle patterns for ball catching. *Frontiers in computational neuroscience*, 7:107, 2013.

[21] A. d'Avella, L. Fernandez, A. Portone, and F. Lacquaniti. Modulation of phasic and tonic muscle synergies with reaching direction and speed. *Journal of neurophysiology*, 100(3):1433–54, 2008.

[22] A. d'Avella and F. Lacquaniti. Control of reaching movements by muscle synergy combinations. *Frontiers in computational neuroscience*, 7:42, 2013.

[23] A. d'Avella, A. Portone, L. Fernandez, and F. Lacquaniti. Control of fast-reaching movements by muscle synergy combinations. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 26(30):7791–810, 2006.

[24] A. d'Avella, P. Saltiel, and E. Bizzi. Combinations of muscle synergies in the construction of a natural motor behavior. *Nature neuroscience*, 6(3):300–8, 2003.

[25] C. De Marchis, M. Schmid, D. Bibbo, A. M. Castronovo, T. D'Alessio, and S. Conforto. Feedback of mechanical effectiveness induces adaptations in motor modules during cycling. *Frontiers in computational neuroscience*, 7:35, 2013.

[26] A. de Rugy, G. E. Loeb, and T. J. Carroll. Are muscle synergies useful for neural control? *Frontiers in computational neuroscience*, 7:19, 2013.

[27] I. Delis, B. Berret, T. Pozzo, and S. Panzeri. Quantitative evaluation of muscle synergy models: a single-trial task decoding approach. *Frontiers in computational neuroscience*, 7:8, 2013.

[28] I. Delis, S. Panzeri, T. Pozzo, and B. Berret. Task-discriminative space-by-time factorization of muscle activity. *Frontiers in Human Neuroscience*, 9:399, 2015.

[29] S. M. S. F. Freitas, M. Duarte, and M. L. Latash. Two kinematic synergies in voluntary whole-body movements during standing. *Journal of neurophysiology*, 95(2):636–645, 2006.

[30] J. Frère and F. Hug. Between-subject variability of muscle synergies during a complex motor skill. *Frontiers in computational neuroscience*, 6:99, 2012.

[31] J. C. Galloway and G. F. Koshland. General coordination of shoulder, elbow and wrist dynamics during multijoint arm movements. *Experimental brain research*, 142(2):163–80, 2002.

[32] B. A. Garner and M. G. Pandy. A Kinematic Model of the Upper Limb Based on the Visible Human Project (VHP) Image Dataset. *Computer methods in biomechanics and biomedical engineering*, 2(2):107–124, 1999.

[33] B. A. Garner and M. G. Pandy. Musculoskeletal model of the upper limb based on the visible human male dataset. *Computer methods in biomechanics and biomedical engineering*, 4(2):93–126, 2001.

[34] R. Gentner, T. Edmunds, D. K. Pai, and A. d'Avella. Robustness of muscle synergies during visuomotor adaptation. *Frontiers in computational neuroscience*, 7:120, 2013.

[35] S. F. Giszter, F. A. Mussa-Ivaldi, and E. Bizzi. Convergent force fields organized in the frog's spinal cord. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 13(2):467–91, 1993.

[36] D. F. Haeufle, M. Günther, A. Bayer, and S. Schmitt. Hill-type muscle model with serial damping and eccentric force-velocity relation. *Journal of biomechanics*, 47(6):1531–6, 2014.

[37] R. Happee and F. C. Van der Helm. The control of shoulder muscles during goal directed movements, an inverse dynamic analysis. *Journal of biomechanics*, 28(10):1179–1191, 1995.

[38] C. B. Hart and S. F. Giszter. A Neural Basis for Motor Primitives in the Spinal Cord. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 30(4):1322–1336, 2010.

[39] S. Heitmann, N. Ferns, and M. Breakspear. Muscle co-contraction modulates damping and joint stability in a three-link biomechanical limb. *Frontiers in neurorobotics*, 5:5, 2011.

[40] N. Hogan. An organizing principle for a class of voluntary movements. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 4(11):2745–54, 1984.

[41] K. R. Holzbaur, W. M. Murray, and S. L. Delp. A Model of the Upper Extremity for Simulating Musculoskeletal Surgery and Analyzing Neuromuscular Control. *Annals of Biomedical Engineering*, 33(6):829–40, 2005.

[42] F. Hug, N. A. Turpin, A. Couturier, and S. Dorel. Consistency of muscle synergies during pedaling across different mechanical constraints. *Journal of neurophysiology*, 106(1):91–103, 2011.

[43] Y. P. Ivanenko, R. E. Poppele, and F. Lacquaniti. Motor control programs and walking. *The Neuroscientist : a review journal bringing neurobiology, neurology and psychiatry*, 12(4):339–48, 2006.

[44] J. Izawa, T. Kondo, and K. Ito. Motor learning model using reinforcement learning with neural internal model. volume 3, pages 3146–51. IEEE, 2003.

[45] S. Jo. A computational neuromusculoskeletal model of human arm movements. *International Journal of Control, Automation and Systems*, 9(5):913–23, 2011.

[46] M. Katayama and M. Kawato. Virtual trajectory and stiffness ellipse during multijoint arm movement predicted by neural inverse models. *Biological cybernetics*, 69(5-6):353–62, 1993.

[47] D. A. Kistemaker, A. J. Van Soest, and M. F. Bobbert. Is equilibrium point control feasible for fast goal-directed single-joint movements? *Journal of neurophysiology*, 95(5):2898–912, 2006.

[48] J. J. Kutch and F. J. Valero-Cuevas. Challenges and New Approaches to Proving the Existence of Muscle Synergies of Neural Origin. *PLoS Computational Biology*, 8(5):e1002434, 2012.

[49] M. A. Lemay and P. E. Crago. A dynamic model for simulating movements of the elbow, forearm, an wrist. *Journal of biomechanics*, 29(10):1319–30, 1996.

[50] C. B. Moody, A. A. Barhorst, and L. Schovanec. A Neuro-Muscular Elasto-Dynamic Model of the Human Arm Part 2: Musculotendon Dynamics and Related Stress Effects. *Journal of Bionic Engineering*, 6(2):108–19, 2009.

[51] F. A. Mussa-Ivaldi, S. F. Giszter, and E. Bizzi. Linear combinations of primitives in vertebrate motor control. *Proceedings of the National Academy of Sciences of the United States of America*, 91(16):7534–38, 1994.

[52] E. J. Nijhof and E. Kouwenhoven. Simulation of multijoint arm movements. In *Biomechanics and Neural Control of Posture and Movement*, chapter 28, pages 363–372. Springer, 2000.

[53] F. Nori and R. Frezza. A control theory approach to the analysis and synthesis of the experimentally observed motion primitives. *Biological cybernetics*, 93(5):323–42, 2005.

[54] S. A. Overduin, A. d'Avella, J. Roh, and E. Bizzi. Modulation of muscle synergy recruitment in primate grasping. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 28(4):880–92, 2008.

[55] E. Pennestrì, R. Stefanelli, P. P. Valentini, and L. Vita. Virtual musculo-skeletal model for the biomechanical analysis of the upper limb. *Journal of biomechanics*, 40(6):1350–61, 2007.

[56] M. Russo, M. D'Andola, A. Portone, F. Lacquaniti, and A. d'Avella. Dimensionality of joint torques and muscle patterns for reaching. *Frontiers in computational neuroscience*, 8:24, 2014.

[57] M. Santello. Kinematic synergies for the control of hand shape. *Archives italiennes de biologie*, 140(3):221–8, 2002.

[58] M. Sartori, L. Gizzi, D. G. Lloyd, and D. Farina. A musculoskeletal model of human locomotion driven by a low dimensional set of impulsive excitation primitives. *Frontiers in computational neuroscience*, 7:79, 2013.

[59] R. Shadmehr and Wise S. P. *The Computational Neurobiology of Reaching and Pointing: A Foundation for Motor Learning*. MIT Press, 2005.

[60] M. Tagliabue, A. L. Ciancio, T. Brochier, S. Eskiizmirliler, and M. A. Maier. Differences between kinematic synergies and muscle synergies during two-digit grasping. *Frontiers in human neuroscience*, 9:165, 2015.

[61] L. H. Ting and S. A. Chvatal. Decomposing Muscle Activity in Motor Tasks: Methods and Interpretation. In F. Danion and M.L. Latash, editors, *Motor Control: Theories, Experiments, and Applications*, pages 102–138. Oxford University Press, 2010.

[62] G. Torres-Oviedo, J. M. Macpherson, and L. H. Ting. Muscle Synergy Organization Is Robust Across a Variety of Postural Perturbations. *Journal of neurophysiology*, 96(3):1530–46, 2006.

[63] G. Torres-Oviedo and L. H. Ting. Subject-specific muscle synergies in human balance control are consistent across different biomechanical contexts. *Journal of neurophysiology*, 103(6):3084–98, 2010.

[64] G. A. Tsianos and G. E. Loeb. Muscle Physiology and Modeling. *Scholarpedia*, 8(10):12388, 2013.

[65] H.E. Veeger, B. Yu, K.N. An, and R.H. Rozendal. Parameters for modeling the upper extremity. 30(6):647–52, 1997.

[66] S. Vernazza-Martin, N. Martin, and J. Massion. Kinematic synergies and equilibrium control during trunk movement under loaded and unloaded conditions. *Experimental brain research*, 128(4):517–26, 1999.

[67] E. R. Westervelt, J. W. Grizzle, C. Chevallereau, J. H. Choi, and B. Morris. *Feedback Control of Dynamic Bipedal Robot Locomotion*. CRC Press, 2007.

[68] D. A. Winter. *Biomechanics and Motor Control of Human Movement*. Wiley, 2009.

[69] J. M. Winters. Hill-based muscle models: a systems engineering perspective. In J. M. Winters and S. L-Y. Woo, editors, *Multiple Muscle Systems*, chapter 5, pages 69–93. Springer, 1990.

[70] J. M. Winters and L. Stark. Estimated mechanical properties of synergistic muscles involved in movements of a variety of human joints. *Journal of biomechanics*, 21(12):1027–41, 1988.

[71] J. E. Wood, S. G. Meek, and S. C. Jacobsen. Quantitation of human shoulder anatomy for prosthetic arm control-I. Surface modelling. *Journal of biomechanics*, 22(3):273–92, 1989.

[72] J. E. Wood, S. G. Meek, and S. C. Jacobsen. Quantitation of human shoulder anatomy for prosthetic arm control-II. Anatomy matrices. *Journal of biomechanics*, 22(4):309–25, 1989.

[73] I. Zelman, M. Titon, Y. Yekutieli, S. Hanassy, B. Hochner, and T. Flash. Kinematic decomposition and classification of octopus arm movements. *Frontiers in computational neuroscience*, 7:60, 2013.

Appendix A

Lumping procedure

Muscles are lumped according to their torque contribution and depending on which joint they span. Briefly, muscles which share the following properties are lumped together and are assumed to be recruited simultaneously:

- they span the same joints;
- are either all monoarticular muscles, or all biarticular;
- they contribute to joint torque in the same direction.

Muscles data from Nijhof and Kouwenhoven [52] are shown in Table A.1 and grouped depending on the lumped muscle they contribute to. Origin and insertion points in red in Table A.1 were missing from [52] and needed to be estimated (see appendix B for more details). The torque contributions to the shoulder (τ_s) and to the elbow (τ_e) could be calculated from the physiological cross-sectional area (PCSA) and the lever arms r_s and r_e (which were assumed to be constant) as follows:

$$\tau_s = k_s \cdot PCSA \cdot r_s \qquad \tau_e = k_e \cdot PCSA \cdot r_e \tag{A.1}$$

where $k_s = 50N/cm^2$ and $k_s = 90N/cm^2$ are the muscle-strength/PCSA ratios (values from [52]). The torque contribution of a muscle is then used as weighting factor to lump all other muscle properties (See [52] for the full details on the lumping procedure).

Table A.1: Anatomical data of the 19 muscles mainly contributing to the upper arm and forearm movements in the horizontal plane. Origins ($O_{(x,y,z)}$), insertions ($I_{(x,y,z)}$), lever arms r_s and r_e and PCSA are taken from [52]. Torques contributions τ_s and τ_e are calculated by means of the procedure proposed in [52]. Positive lever arms/torques represent flexions, negative lever arms/torques give raise to extensions. Reference system for origin and insertion points is depicted in Figure 2.4 in section 2.2.4.

Muscle	O_x	O_y	O_z	I_x	I_y	I_z	r_s	r_e	PCSA	τ_s (Nm)	τ_e (Nm)
SF - Shoulder Flexor											
DELC[1]	-3.4	2.2	2.8	12.4	1.0	1.4	3.9		4.5	8.78	
PMJA[2]	**4.6**	**18.6**	**-12.6**	6.3	1.7	1.1	4.8		3.9	9.36	
PMJC[3]	-11.8	7.6	0.0	9.4	1.7	0.9	8.2		5.2	21.32	
PMJS[4]	**-4.4**	**16.0**	**-12.2**	6.9	1.7	1.1	6.6		4.5	14.85	
SE - Shoulder Extensor											
DELA[5]	1.0	-1.5	2.3	15.3	0.6	1.3	-1.0		13.5	-6.75	
DELS[6]	-4.7	-6.0	0.8	13.2	-0.1	1.2	-6.1		3.9	-11.90	
LATD[7]	**13.5**	**-4.8**	**-18.6**	5.5	0.9	-1.2	-1.9		12.9	-12.26	
TMAJ[8]	-4.0	-8.3	-8.9	6.2	0.4	-1.0	-4.2		5.8	-12.18	
EF - Elbow Flexor											
BRAD[9]	23.0	-1.1	0.8	29.1	22.6	-2.8		4.2	1.5		5.67
BRAC[10]	16.1	0.0	0.1	31.8	3.5	-1.3		2.1	7.0		13.23
PROT[11]	28.7	-0.5	0.4	30.6	7.5	0.1		1.6	3.4		4.90
ECRD[12]	27.8	-1.1	1.4	**31.0**	**24.0**	**0.0**		2.9	5.3		13.83
EF - Elbow Extensor											
ANCO[13]	30.6	-1.1	1.0	33.0	0.8	-0.8		-1.1	2.5		-2.48
ECUL[14]	30.3	-1.4	1.7	**31.0**	**24.0**	**0.0**		-1.3	3.4		-3.98
TRIA[15]	13.8	-1.9	0.8	32.6	-1.9	-1.1		-2.0	6.0		-10.80
TRIM[16]	20.5	-1.7	-1.9	32.6	-1.9	-1.1		-2.0	6.1		-10.98
BF - Biarticular Flexor											
BILH[17]	-1.1	1.9	1.3	31.0	4.4	-0.1	1.9	3.4	2.5	2.38	7.65
BISH[18]	-3.2	3.3	0.8	31.0	4.4	-0.1	4.6	3.4	2.1	4.83	6.43
BE - Biarticular Extensor											
TRIO[19]	-1.4	-2.8	-2.8	32.6	-1.9	-1.1	-3.2	-2.0	6.7	-10.72	-12.06

[1] deltoid - clavicular part [2] pectoralis major - abdominal part [3] pectoralis major - clavicular part
[4] pectoralis major - sterno-costal part [5] deltoid - acromial part [6] deltoid - scapular part
[7] latissimus dorsi [8] teres major [9] brachioradialis [10] brachialis [11] pronator teres [12] extensor carpi radialis [13] anconeus
[14] extensor carpi ulnaris [15] triceps-lateralis [16] triceps-medialis [17] biceps-long head [18] biceps-short head [19] triceps-longum

Appendix B

Missing musculoskeletal data

Insertion and origin coordinates marked in red in table A.1 are missing in [52] because either their insertion or their origin point are not on the bone segments considered. Nevertheless they contribute to the joint torques and therefore they had to be found in order to be included into our model.

Shoulder data Data in [52] for the shoulder is taken from [72] where instead the coordinates for the muscles PMJA, PMJS and LATD are all available. Note however that in [52] the authors have applied a number of transformations to the original data in part using a skeleton as model. To make a gross approximation we have matched all the available data from [72] with all the available data from [52] and have estimated the Rotation between the two sets using the well known Kabsch algorithm. The translation was instead estimated by the distance between the centroids of the two datasets after applying the rotation from Kabsch. Estimated rotation and translation are shown in (B.1). Numbers are expresses in meters.

$$\boldsymbol{R} = \begin{pmatrix} -0.471017 & -0.777509 & -0.416681 \\ -0.617529 & -0.046682 & 0.785162 \\ -0.629922 & 0.627137 & -0.458146 \end{pmatrix} \qquad \boldsymbol{t} = \begin{pmatrix} 0.45937 \\ -0.41399 \\ 0.38223 \end{pmatrix} \quad \text{(B.1)}$$

Elbow data Unfortunately there was no data available for the muscles ECRD and ECUL. These two muscles originate close to the elbow joint and insert into the wrist/hand complex. We have done a gross approximation by considering the muscle lines of actions shown in [52].

Appendix C

Including dynamical system boundaries in DRD

In this appendix we formulate a method to take into account dynamical system variables boundaries into the DRD (Dynamic Response Decomposition) framework, described in section 2.1. The method was used in this work to generate feasible trajectories for the musculoskeletal model described in 2.2.

C.1 Introduction

Given a dynamical system (e.g. a kinematic chain), its dynamic responses (e.g. joint trajectories) defined as in Eq. (2.4) and a task defined by constraints (e.g. a reaching task) as in Eq. (2.5), the problem of finding a trajectory by interpolation on the dynamic responses, which satisfies the task constraints, is formalized as in Eq. (2.8). For convenience we report Eq. (2.8):

$$\boldsymbol{M}\boldsymbol{a} = \boldsymbol{P} \tag{C.1}$$

where $\boldsymbol{P}$ is the vector of task constraints and $\boldsymbol{M}$ is the alternant matrix, containing the dynamic responses evaluated at the timestamps where the task constraints are defined. The problem, therefore, reduces to find the vector $\boldsymbol{a}$ which solves Eq. (C.1).

The actual implementation of DRD, does not include the possibility to define boundaries to the obtained solutions, in terms of lower bound and upper bound to the values taken by the dynamical system state variables and their derivatives. Such boundaries might either derive by the nature of the dynamical system, or they might be imposed by the engineer. In the following we will refer to these boundaries simply as *dynamical system boundaries*.

As an example, DRD was used in previous works to generate synergies for a 2-dof kinematic chain actuated in torque space. It was not possible however to specify directly the joint angular boundaries. This resulted in solutions to Eq. (C.1) leading to trajectories which could potentially exceed the dynamical system boundaries. The user of DRD had to work in a trial and error way, that is:

1. Search with DRD for a solution to the task in the phase space (e.g. the kinematic space of the kinematic chain) by inverting the constraints equation as follows:
$$\boldsymbol{a} = \boldsymbol{M}^{-1}\boldsymbol{P} \tag{C.2}$$
Most of the times the system of equations (C.1) is overdetermined or underdetermined, therefore the Moore–Penrose pseudoinverse of $\boldsymbol{M}$, which existence and uniqueness is guaranteed, is used. We will call therefore this method the *pseudoinverse* method or approach.

2. Verify a posteriori if the solution is inside the boundaries using some user-defined method. If not, discard it and start some other search to find a linear combination of dynamic responses solving the task. In previous work this was done by randomizing the linear combination coefficients. This search stops when a solution of Eq. (C.1) with negligible error and entirely inside the boundaries has been found or when a maximum number of iterations have been reached. This is a time-consuming step and it does not guarantee to find a solution within the boundaries. We will call this method the *randomized* method or approach.

Solutions given by the pseudoinverse approach inverting Eq. (C.1) can potentially exceed the dynamical system boundaries. Objective of the method it to include boundaries in the formulation of DRD, thus enabling to find solutions to Eq. (C.1) which a priori lay entirely within the boundaries.

C.2 Method development

Given a task defined by $\boldsymbol{P}$, and assuming that a solution to Eq. (C.1) exists, such a solution may exceed the imposed boundaries. What we need is to find a solution to Eq. (C.1) which is inside the boundaries and, if possible, close to the ideal solution given by Eq. (C.2). The goodness of a solution is measured by the interpolation error err_I (see section 2.1.4), which is the distance between the solution and the task constraints. We want to find the solution which minimizes the interpolation error, subject to

the dynamical system boundaries constraints. We will now formulate this problem mathematically. From the definition of interpolation error err_I for k task constraints on an n-dimensional dynamical system we have (see section 2.1.4):

$$err_I = \sqrt{\sum_{k \in K} err_{I_k}^2} = \sqrt{\sum_{k \in K} ||\boldsymbol{q}_k - \boldsymbol{\theta}_k \boldsymbol{a}||^2} \tag{C.3}$$

$$err_I^2 = \sum_{k \in K} ||\boldsymbol{q}_k - \boldsymbol{\theta}_k \boldsymbol{a}||^2 = \sum_{k \in K} \sum_{j=1}^{n} (q_{kj} - \boldsymbol{\theta}_{kj} \boldsymbol{a})^2 \tag{C.4}$$

where $\boldsymbol{q}_k = [q_{k1}, q_{k2} ... q_{kn}]^T$ is a subvector of $\boldsymbol{P}$ and q_{kj} is the scalar value of the constraint k at dimension j (e.g. joint), $\boldsymbol{\theta}_k = [\boldsymbol{\theta}_{k1}, \boldsymbol{\theta}_{k2} ... \boldsymbol{\theta}_{kn}]^T$ is a submatrix of $\boldsymbol{M}$ and $\boldsymbol{\theta}_{kj}$ is the *1*-by-m vector of constrained dynamic responses values at constraint k and dimension j, with m the number of dynamic responses, equal to the number of columns of $\boldsymbol{M}$. Developing the expression of err_I^2 we have:

$$\begin{aligned} err_I^2 &= \sum_{k \in K} \sum_{j=1}^{n} (q_{kj}^2 + (\boldsymbol{\theta}_{kj} \boldsymbol{a})^2 - 2 q_{kj} \boldsymbol{\theta}_{kj} \boldsymbol{a}) \\ &= \sum_{k \in K} (\sum_{j=1}^{n} q_{kj}^2 + \sum_{j=1}^{n} (\boldsymbol{\theta}_{kj} \boldsymbol{a})^2 - 2 \sum_{j=1}^{n} q_{kj} \boldsymbol{\theta}_{kj} \boldsymbol{a}) \\ &= \sum_{k \in K} (\boldsymbol{q}_k^T \boldsymbol{q}_k + (\boldsymbol{\theta}_k \boldsymbol{a})^T (\boldsymbol{\theta}_k \boldsymbol{a}) - 2 \boldsymbol{q}_k^T \boldsymbol{\theta}_k \boldsymbol{a}) \\ &= \sum_{k \in K} \boldsymbol{q}_k^T \boldsymbol{q}_k + \sum_{k \in K} \boldsymbol{a}^T \boldsymbol{\theta}_k^T \boldsymbol{\theta}_k \boldsymbol{a} - 2 \sum_{k \in K} \boldsymbol{q}_k^T \boldsymbol{\theta}_k \boldsymbol{a} \end{aligned} \tag{C.5}$$

Given that $\sum_{k \in K} \boldsymbol{q}_k^T \boldsymbol{q}_k = \boldsymbol{P}^T \boldsymbol{P}$, $\sum_{k \in K} \boldsymbol{\theta}_k^T \boldsymbol{\theta}_k = \boldsymbol{M}^T \boldsymbol{M}$ and $\sum_{k \in K} \boldsymbol{q}_k^T \boldsymbol{\theta}_k = \boldsymbol{P}^T \boldsymbol{M}$ the equation can be rewritten as

$$err_I^2 = \boldsymbol{P}^T \boldsymbol{P} + \boldsymbol{a}^T \boldsymbol{M}^T \boldsymbol{M} \boldsymbol{a} - 2 \boldsymbol{P}^T \boldsymbol{M} \boldsymbol{a} \tag{C.6}$$

indeed, if $\tilde{\boldsymbol{P}}$ would solve Eq. (C.1) for some $\boldsymbol{a}$, the squared error would be:

$$err_I^2 = \boldsymbol{P}^T \boldsymbol{P} + \tilde{\boldsymbol{P}}^T \tilde{\boldsymbol{P}} - 2 \boldsymbol{P}^T \tilde{\boldsymbol{P}} = (\boldsymbol{P} - \tilde{\boldsymbol{P}})^2 \underset{\boldsymbol{P} \to \tilde{\boldsymbol{P}}}{\longrightarrow} 0 \tag{C.7}$$

Our objective is to find $\boldsymbol{a}$ which minimizes the interpolation error, that is:

$$\begin{aligned}\boldsymbol{a} &= \underset{\boldsymbol{a}}{\operatorname{argmin}}(err_I) = \underset{\boldsymbol{a}}{\operatorname{argmin}}(err_I^2) \\ &= \underset{\boldsymbol{a}}{\operatorname{argmin}}(\boldsymbol{P}^T\boldsymbol{P} + \boldsymbol{a}^T\boldsymbol{M}^T\boldsymbol{M}\boldsymbol{a} - 2\boldsymbol{P}^T\boldsymbol{M}\boldsymbol{a}) \\ &= \underset{\boldsymbol{a}}{\operatorname{argmin}}(\frac{1}{2}\boldsymbol{P}^T\boldsymbol{P} + \frac{1}{2}\boldsymbol{a}^T\boldsymbol{M}^T\boldsymbol{M}\boldsymbol{a} - \boldsymbol{P}^T\boldsymbol{M}\boldsymbol{a}) \\ &= \underset{\boldsymbol{a}}{\operatorname{argmin}}(\frac{1}{2}\boldsymbol{a}^T\boldsymbol{M}^T\boldsymbol{M}\boldsymbol{a} - \boldsymbol{P}^T\boldsymbol{M}\boldsymbol{a}) \\ &= \underset{\boldsymbol{a}}{\operatorname{argmin}}(f(\boldsymbol{a}))\end{aligned} \tag{C.8}$$

which leads to the standard formulation of a Quadratic Programming problem with objective function:

$$\begin{aligned}&f(\boldsymbol{a}) = \frac{1}{2}\boldsymbol{a}^T\boldsymbol{Q}\boldsymbol{a} + \boldsymbol{c}^T\boldsymbol{a} \\ &\text{with:} \\ &\boldsymbol{Q} = \boldsymbol{M}^T\boldsymbol{M} \\ &\boldsymbol{c} = -\boldsymbol{M}^T\boldsymbol{P}\end{aligned} \tag{C.9}$$

From the mathematical and geometrical interpretation of $f(\boldsymbol{a})$ we conclude that:

1. $\boldsymbol{Q}$ is the Gram Matrix of $\boldsymbol{M}$ and is therefore positive semidefinite and symmetric by definition.

2. The term $\frac{1}{2}\boldsymbol{a}^T\boldsymbol{Q}\boldsymbol{a}$ is a quadratic form. Because $\boldsymbol{Q}$ is positive semidefinite, its eigenvalues are non negative and the quadratic form is positive semidefinite, that is, it has a single global minimum in $\boldsymbol{a} = \boldsymbol{0}$.

3. The term $\boldsymbol{c}^T\boldsymbol{a}$ is an affine hyperplane which simply shifts the paraboloid $\frac{1}{2}\boldsymbol{a}^T\boldsymbol{Q}\boldsymbol{a}$.

Figure C.1 shows an example plot of $f(\boldsymbol{a})$ for a 2-dimensional $\boldsymbol{a}$.

Thus, if the problem is feasible (i.e. the task constraints $\boldsymbol{P}$ lay within the boundaries), solving the Quadratic Programming problem with objective function (C.9) subject to the dynamical system boundaries guarantees to find the global optimal solution to Eq. (C.1). We will call this method the *optimization* method or approach.

We will now demonstrate that solutions found by minimizing (C.9) without being subject to boundary constraints cannot have a greater interpolation error than the solutions found with the pseudoinverse method. The Moore-Penrose pseudoinverse of a matrix is known to exists and to be unique

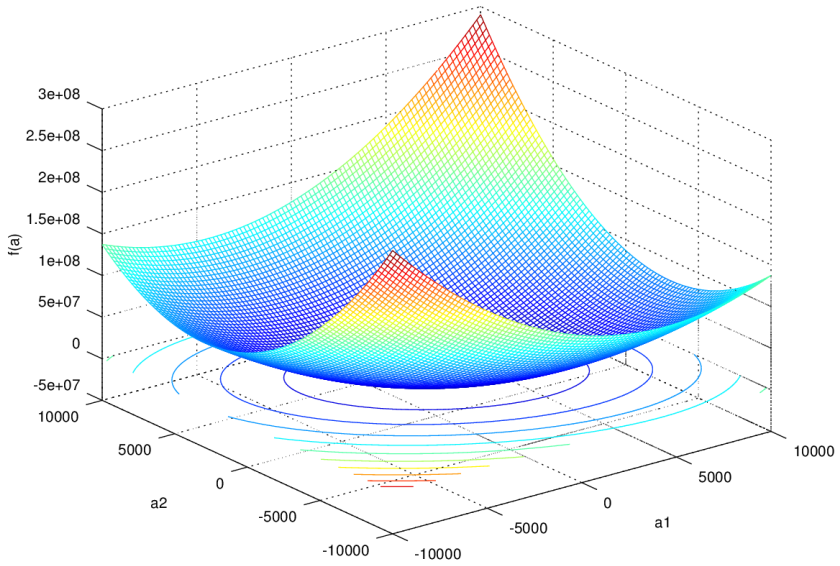


Figure C.1: Example of the quadratic programming objective function used to solve Eq. (C.1) *subject to the dynamical system boundaries.*

for every matrix and we have shown that $f(\boldsymbol{a})$ has a global minimum. The interpolation error above can be rewritten as follows:

$$err_I = ||\boldsymbol{P} - \boldsymbol{M}\boldsymbol{a}|| \tag{C.10}$$

It is known that the pseudoinverse of a matrix provides a least squares solution to a system of linear equations. The least squares solution minimizes in this case the euclidean norm $||\boldsymbol{P} - \boldsymbol{M}\boldsymbol{a}||$, meaning that

$$\boldsymbol{a} = \boldsymbol{M}^{+}\boldsymbol{P} = \underset{\boldsymbol{a}}{\operatorname{argmin}}\, err_I = \underset{\boldsymbol{a}}{\operatorname{argmin}}\, f(\boldsymbol{a}) \tag{C.11}$$

The pseudoinverse solution corresponds with the one minimizing $f(\boldsymbol{a})$ and both minimize the interpolation error. This means that for an unconstrained system the two methods are equivalent. However for a system for which boundaries are defined and the pseudoinverse solution lays outside the boundaries, the optimized solution is the closest possible solution to the pseudoinverse one, laying inside the boundaries. This is shown graphically in Figure C.2 for a 1-dimensional example function.

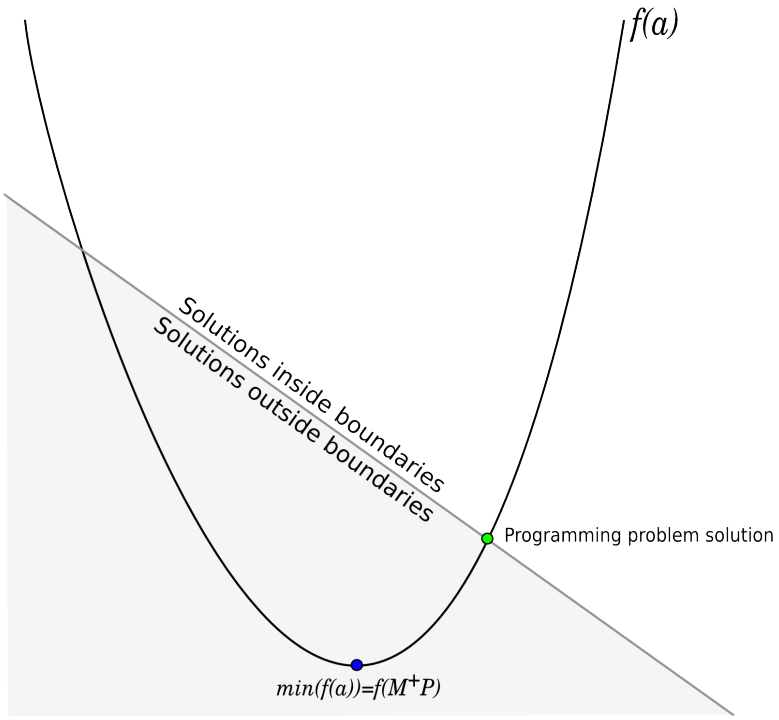


Figure C.2: Geometrical interpretation of the solutions found by pseudoinverse and by optimization.

C.3 Test procedure

Similarly to [2], we test the method on a 2-joint planar kinematic chain representing a human arm model. Anthropometric data (link length, mass, center of mass, moment of inertia and joint boundaries) are taken from

literature [52]. As described in [2] the chain has been stimulated with a number of random actuations (exploration synergies) to obtain the so called exploration set of dynamic responses. For the described kinematic chain the exploration synergies are time-varying joint torques and the dynamic responses are the corresponding joint angle trajectories. The result of the exploration can be viewed in Figure C.3.

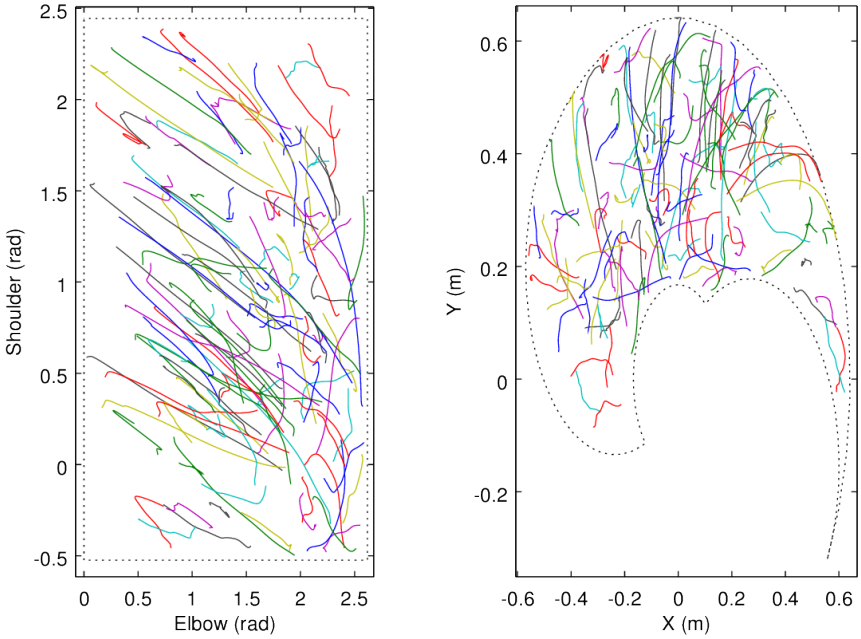


Figure C.3: Dynamic responses exploration set represented in the joint space (left) and in the effector space (right) of the kinematic chain used to test the method. Each colored line is a different dynamic response, composed by the angular joint trajectories of the kinematic chain.

The exploration set has then been used to solve generic point to point reaching tasks.

C.4 Results

In our tests a generic reaching task is defined by know initial and final position, null initial and final velocity and unconstrained acceleration. That

is, Eq. (C.1) can be written as:

$$\boldsymbol{M}\boldsymbol{a} = \begin{pmatrix} \boldsymbol{\theta}_1(0) & ... & \boldsymbol{\theta}_N(0) \\ \boldsymbol{\theta}_1(T) & ... & \boldsymbol{\theta}_N(T) \\ \dot{\boldsymbol{\theta}}_1(0) & ... & \dot{\boldsymbol{\theta}}_N(0) \\ \dot{\boldsymbol{\theta}}_1(T) & ... & \dot{\boldsymbol{\theta}}_N(T) \end{pmatrix} \boldsymbol{a} = \begin{pmatrix} \boldsymbol{q}_0 \\ \boldsymbol{q}_T \\ \dot{\boldsymbol{q}}_0 \\ \dot{\boldsymbol{q}}_T \end{pmatrix} = \boldsymbol{P} \tag{C.12}$$

where $\boldsymbol{\theta}_j(t)$, $\dot{\boldsymbol{\theta}}_j(t)$, $\boldsymbol{q}_j$ and $\dot{\boldsymbol{q}}_j$ are n-dimensional vectors with n the number of dimensions of the dynamical system. In our case $n = 2$, the number of joints. The dynamical system boundaries are defined by the minimum and the maximum angular position of the elbow and shoulder joint. However, because of the generality of the method, boundaries can be defined on any derivate of the system state variables. We say that a given task is feasible if all its constraints lay within the boundaries. We say that a trajectory of the system variables is feasible if it lays entirely within the imposed boundaries. We therefore seek a solution to (C.1) that leads to a feasible trajectory.

For a given task we can identify following cases:

1. The task is not feasible, that is, some constraints lay outside the dynamical system boundaries.

2. The task is feasible and the pseudoinverse approach produces a feasible trajectory with negligible interpolation error.

3. The task is feasible but the pseudoinverse approach does not produce a feasible trajectory.

In the following sections we describe results and comparisons between the pseudoinverse and the optimization approach in all these cases.

Case 1: Task not feasible

In general, finding a solution to the system of equations (C.1) does not mean that the trajectories of the system variables are inside the boundaries. The kinematic chain under test, for example, may not be able to perform a movement because at some point the position trajectory exceeds the joints boundaries. In this section we want to point out that the optimization approach always produces feasible trajectories a priori. As an example, we can imagine an object in front of us which is a bit more distant than the full extension of our arm. We may think to be able to reach it, although we are not. Nevertheless we still try the move toward the object. Figure C.4 shows a comparison between the pseudoinverse and the optimization approach for

feasible (inside the dotted box) and not feasible tasks (outside the dotted box). For the non feasible tasks, the optimization approach still produces feasible trajectories (Figure C.4b), which of course will have a potentially non-negligible interpolation error as they will be inside the joint boundaries. This is exactly what happens when we try to reach an object which is too far and we are finally not able to reach it.

Trajectories found by means of the pseudoinverse approach are instead not feasible (Figure C.4a). Being the pseudoinverse of a matrix unique by definition we conclude that the pseudoinverse approach fails to find a valid trajectory. We will show later that even if the task is feasible, it is not always possible for the pseudoinverse method to find a feasible trajectory.

Case 2: Task is feasible and the pseudoinverse trajectory is feasible

In this case the pseudoinverse approach produces a feasible trajectory. Because of the uniqueness of both $\boldsymbol{M}^+$ and $\min f(\boldsymbol{a})$ the trajectories found by the two methods must necessary be the same. This is shown in Figure C.5 where the trajectories corresponding to the two approaches are perfectly overlapping. The interpolation error err_I for all tasks and both approaches is negligible, at machine precision level (10^{-15}).

Case 3: Task is feasible but the pseudoinverse trajectory is not feasible

This is the most interesting case and the one which justifies the development of the optimization method. If the dynamical system boundaries are tight or the task constraints are close to the boundaries it is possible that the pseudoinverse approach does not find a feasible trajectory. Figure C.6 shows an example. Some trajectories found by the two methods do not correspond anymore and while the pseudoinverse approach (blue dashed lines) fails for several tasks, the optimization approach is still able to find trajectories (green lines) which are inside the boundaries. These have however a greater and perhaps non-negligible interpolation error[1].

[1]We have demonstrated in Eq. (C.11) that $\min f(\boldsymbol{a}) = f(\boldsymbol{M}^+\boldsymbol{P})$. We are minimizing the interpolation error and the smallest error is at the function minimum. But because this is also the solution to Eq. (C.1) provided by pseudoinverse (which leads to an unfeasible trajectory) the interpolation error must be necessary greater. Better results would be achieved only with a bigger exploration set, but then also the pseudoinverse approach would perform better.

The approach in this case, as stated in section C.1, would be to apply the randomized method. As an example, let us consider one of the tasks depicted in Figure C.6b (35th task with filled black circle) for which the pseudoinverse trajectory is not feasible. Figure C.7 compares the trajectories obtained by the pseudoinverse, the randomized and the optimization approaches. The search for randomized trajectories is expensive and even if one is found, it is mostly not acceptable from a kinematic point of view. We will support this last statement with another example.

From a starting position where the elbow is nearly fully extended and the shoulder is about 80% flexed we search trajectories for 9 reaching tasks regularly distributed onto the feasible area. The found trajectories are shown in Figure C.8. Tasks 7 and 9 are solved by means of the randomized method (Figure C.8a) and have awkward trajectories, whereas the optimization approach leads to very smooth trajectories.

Obviously the interpolation errors of the trajectories solving tasks 7 and 9 found by the optimization approach will be higher than those found by the randomized approach. This is shown in Figure C.9. Nevertheless, these are by definition the best possible trajectories one can ever find with the specific dynamic responses set in use. An improvement would be only possible by increasing the number of dynamic responses or by changing the torque profiles used to actuate the kinematic chain. From a kinematic point of view the trajectories found by randomization are much more cumbersome than those found through optimization. Figure C.10 shows a comparison between joint position, velocity and acceleration profiles of the trajectory solving task 7 for the two approaches. Figure C.11 shows the corresponding position trajectories in the end effector space.

Comparison in the end effector space

From the previous sections, for feasible tasks, we can summarize results as follows:

1. If a pseudoinverse trajectory is feasible then the one found by the optimization approach is identical and both have negligible interpolation error;

2. If a pseudoinverse trajectory is not feasible then a randomized trajectory with negligible error is searched:

   (a) if such a trajectory is found it will have a better interpolation error than the trajectory found through optimization, but it will be less smooth;

(b) if such a trajectory cannot be found the task remains unsolved by the pseudoinverse and randomized approaches, while the optimization method will always be able to solve it;

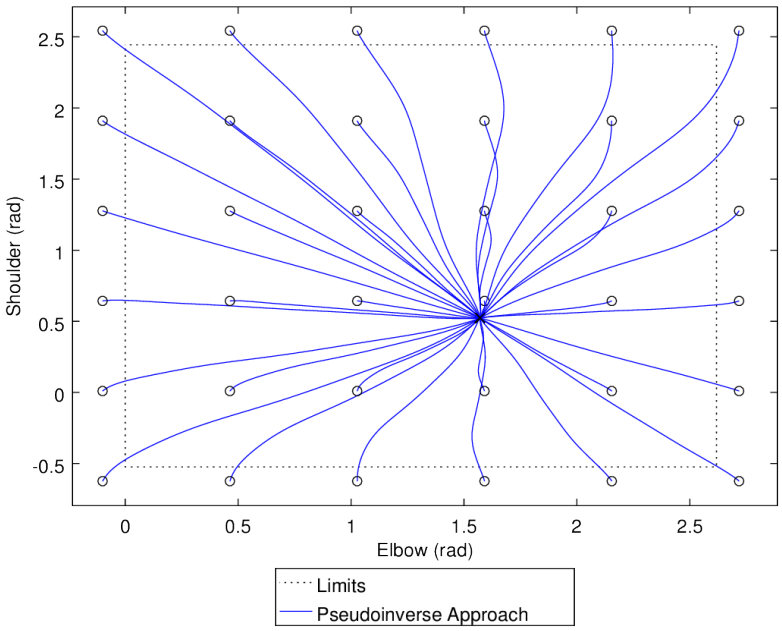
That is, either the trajectories and the interpolation errors for the pseudoinverse and the optimization approaches are the same, or the interpolation errors of the pseudoinverse and randomized approach are smaller but their trajectories are unacceptable or cannot be found.

We show this with another example in the end effector space. For several starting positions we simulated reaching tasks to over 3000 destination positions equally distributed over the space. We searched randomized trajectories when those produced by the pseudoinverse method were not feasible. At each destination position we calculated following data:

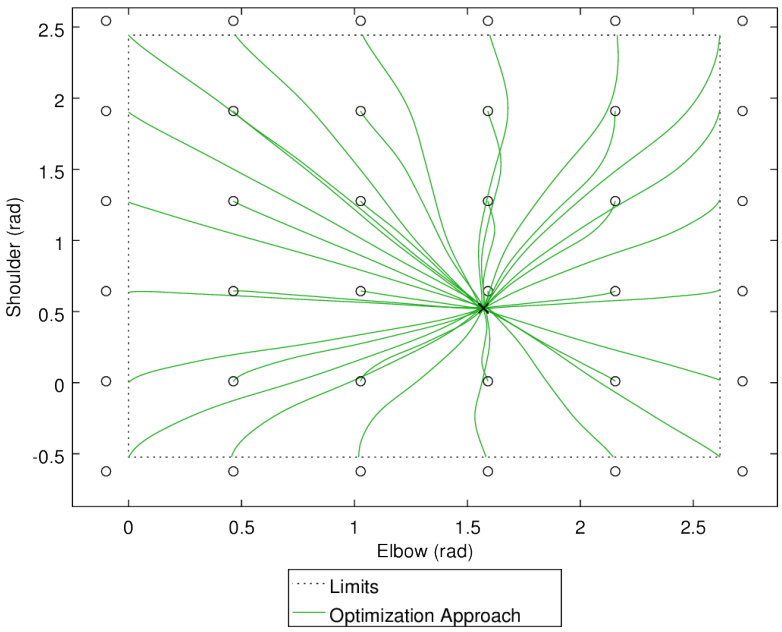
1. maximum difference between interpolation error produced by optimization and the one obtained by pseudoinverse or randomization. This difference is expected to increase when the pseudoinverse trajectory is not feasible and a randomized trajectory with negligible interpolation error have been found. Figure C.12a shows how this happened on the boundary where some points are darker because the interpolation errors produced by the optimization approach were higher.

2. maximum norm of the difference between optimized and pseudoinverse or randomized position trajectories. Figure C.12b shows how this norm increases in the same areas where the interpolation error does. This is because the randomized trajectories, while having negligible interpolation error, contribute more to the norm. The darkest points correspond to destination positions for which neither pseudoinverse nor randomized feasible trajectories could be found and the norm is virtually infinite.

As an example, Table C.1 shows data for several reaching points from one of the starting positions. Reaching destinations are all in the bottom-left area of the end effector space where the most problems with the pseudoinverse approach arise for the chosen starting point (see Figure C.12). The relation between the interpolation errors and the norm of the difference of the trajectories can be clearly seen in the table. Rows with ∞ in the interpolation error and norm columns indicate that for this points no feasible trajectory was found, neither by pseudoinverse, nor by randomization.



(a)



(b)

Figure C.4: Comparison between trajectories solving tasks (circles) found by the pseudoinverse approach (a) and by the optimization approach (b). The trajectories are given in the joint space. The black dotted box identifies the feasible area delimited by the shoulder and elbow joint angular boundaries. The circles outside the box represent non feasible tasks. While for the non feasible tasks the pseudoinverse trajectories are also unfeasible, the optimization approach is still able to produce feasible trajectories.

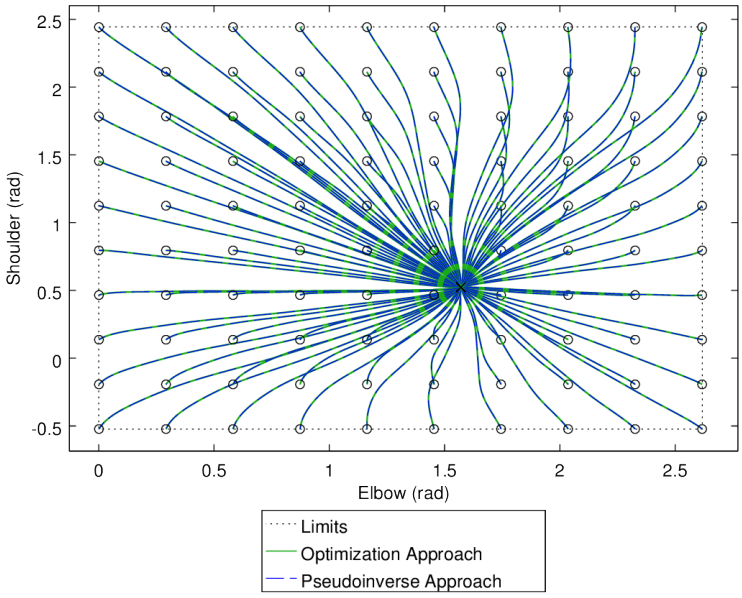
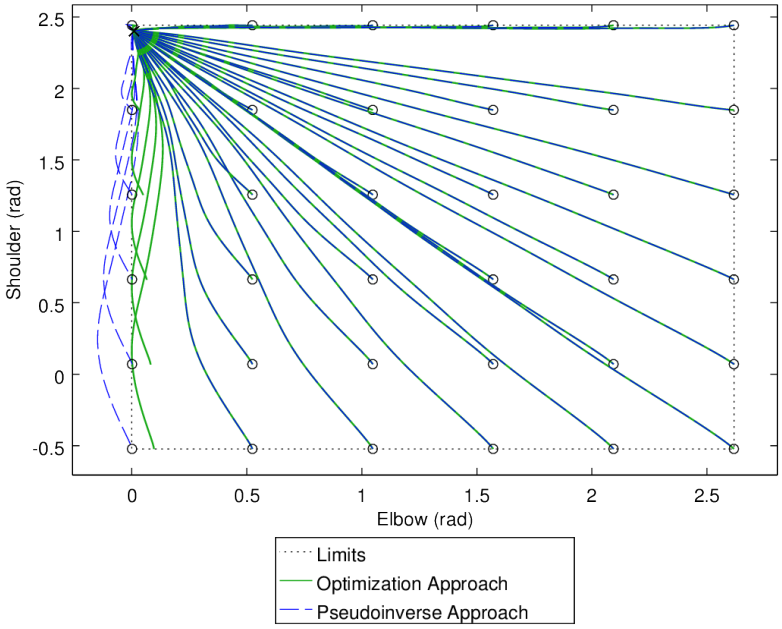
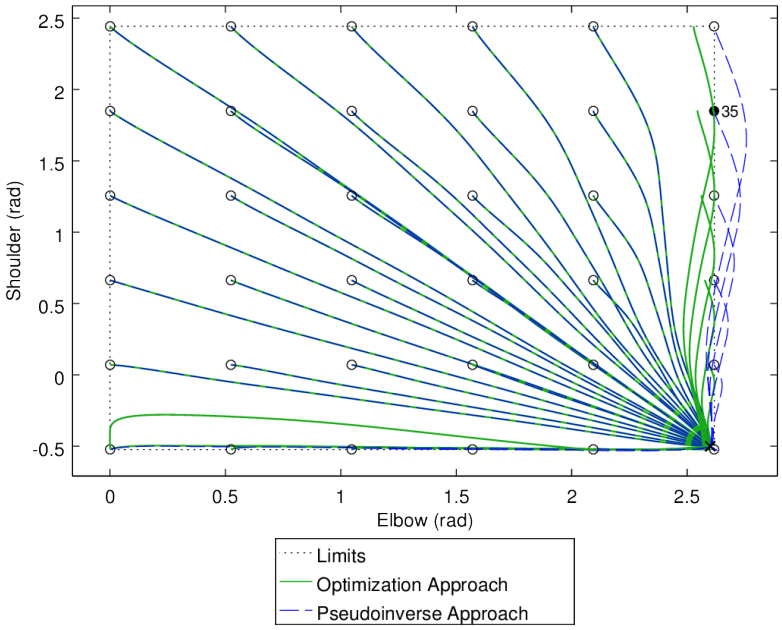


Figure C.5: Comparison between trajectories solving tasks (circles) in joint space found by the pseudoinverse approach (dashed blue trajectories) and the optimization approach (green trajectories). The starting point of all reachings is close to the middle point of the joint boundaries area and is marked with a cross. Trajectories produced by the two approaches are perfectly overlapping.



(a)



(b)

Figure C.6: Comparison between trajectories solving tasks (circles) in joint space found by the pseudoinverse approach (dashed blue trajectories) and the optimization approach (green trajectories) when the starting point of the reaching movements is close to the boundary. The leftmost blue dashed trajectories outside the boundaries in (a) and the rightmost ones in (b) are not feasible.

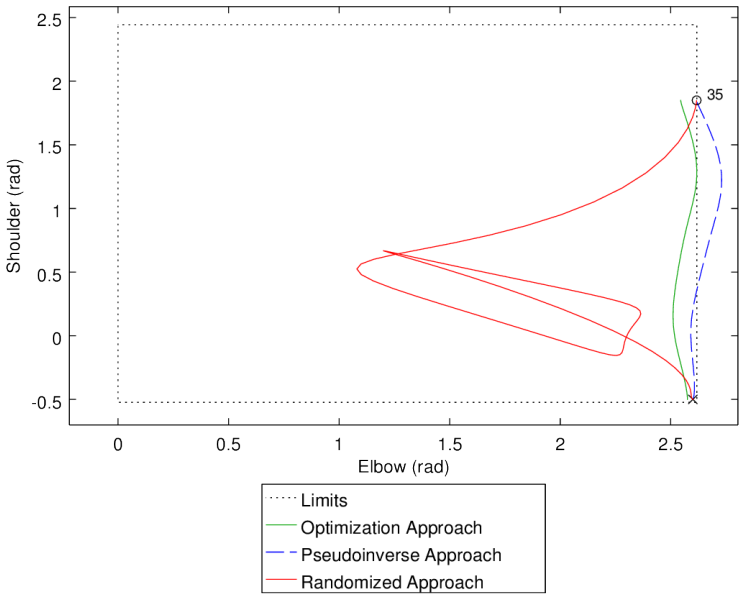
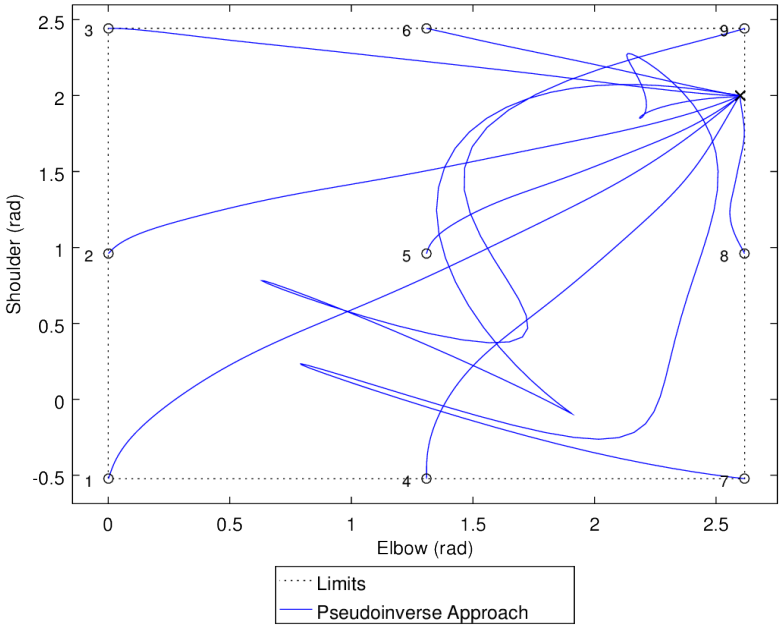
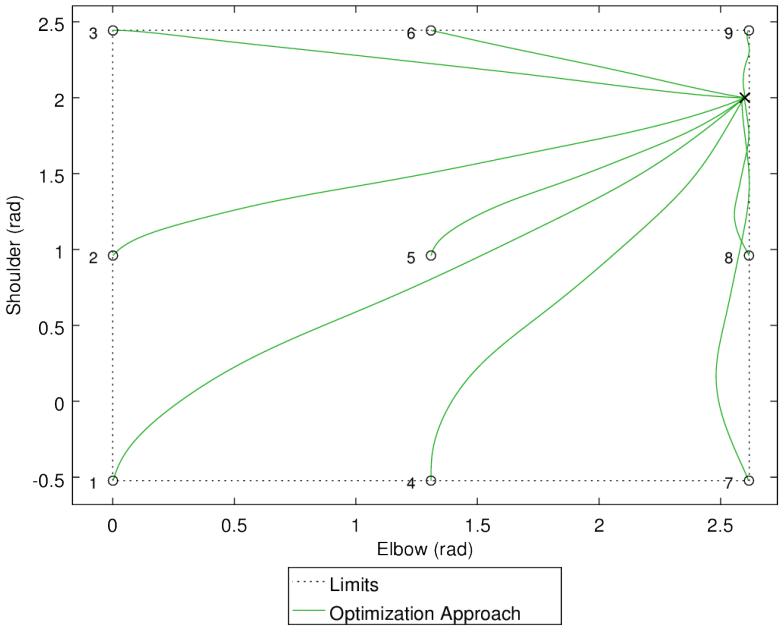


Figure C.7: Example of randomized trajectory when the pseudoinverse approach is not able to find a feasible trajectory for a task. The blue dashed line is the unfeasible trajectory found by pseudoinverse, the green line is the feasible trajectory found by optimizing, the red line is the feasible randomized trajectory.



(a)



(b)

Figure C.8: Comparison between pseudoinverse approach (a) and optimization approach (b) for the solution of reaching tasks from a boundary position to 9 regularly distributed final positions. In (a) tasks 7 and 9 are solved by randomization because the pseudoinverse trajectories are not feasible.

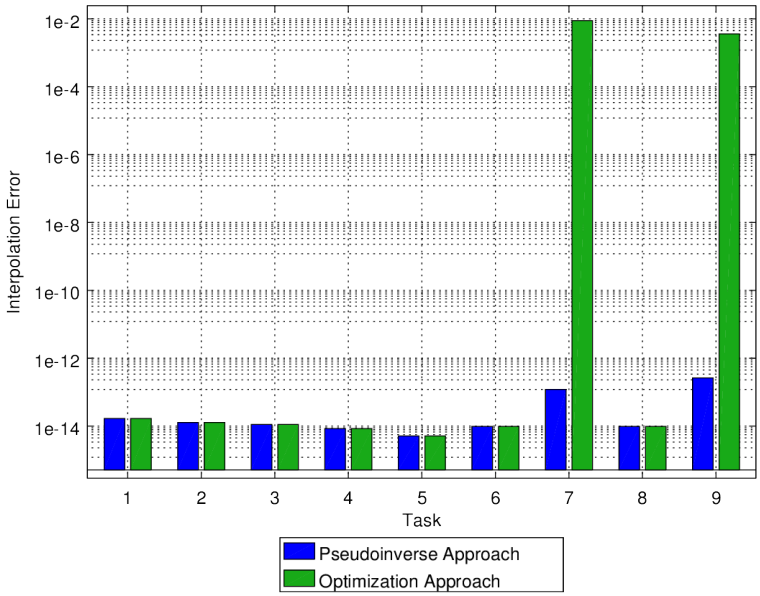
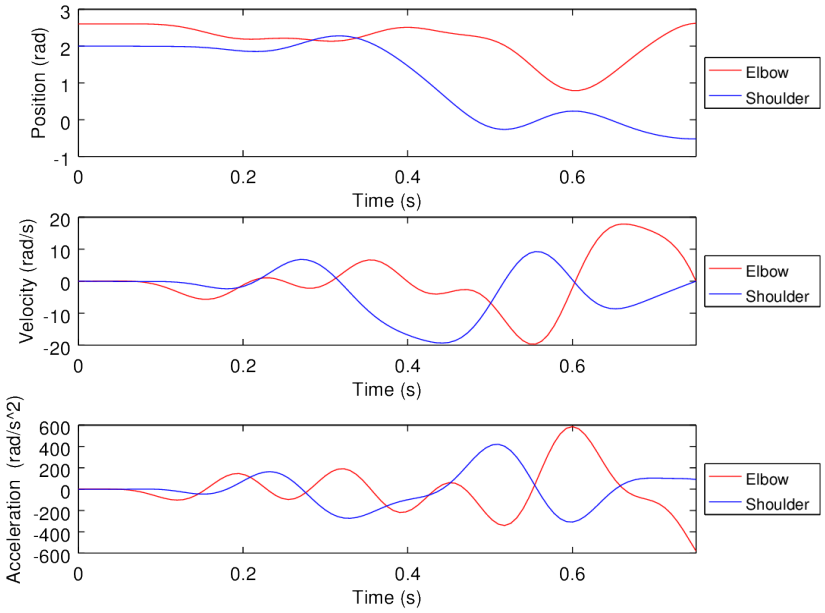
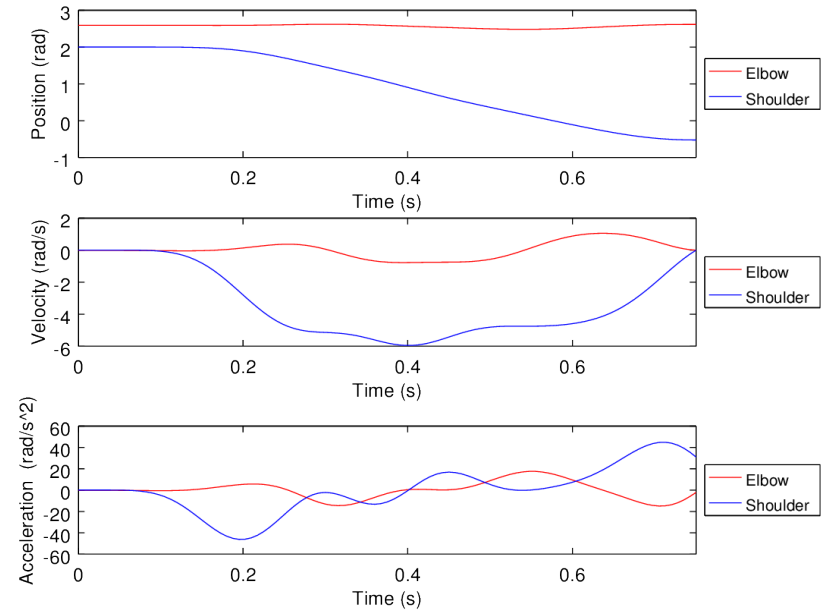


Figure C.9: Comparison between interpolation error of the pseudoinverse and of the optimization approaches for the tasks in Figure C.8.

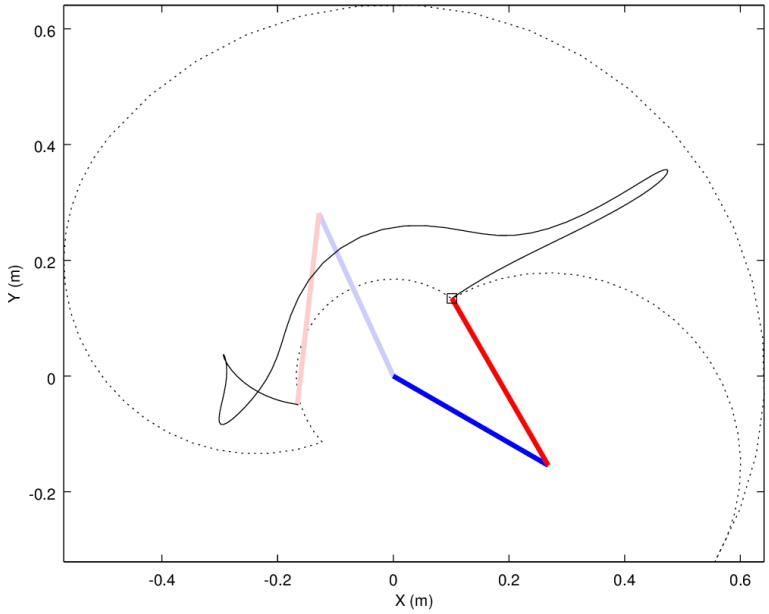


(a)

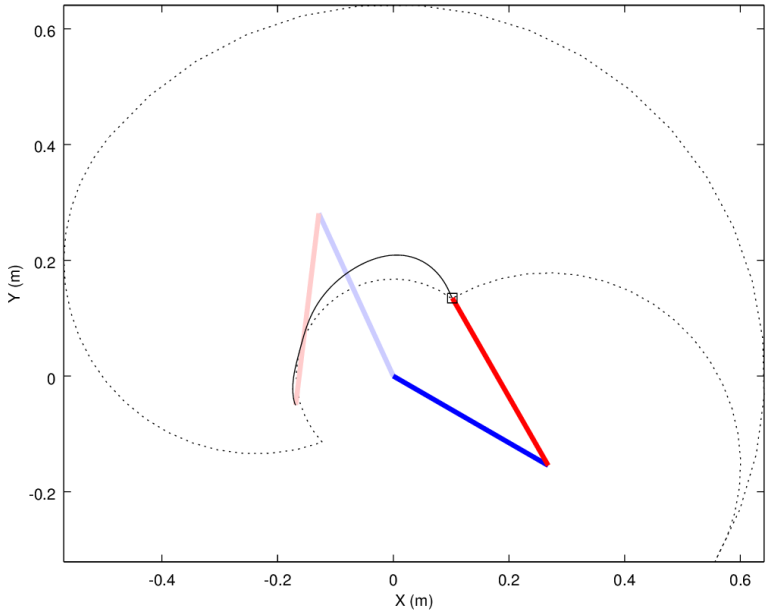


(b)

Figure C.10: Comparison between joint kinematic profiles of the trajectories solving task 7 in Figure C.8. (a) position, velocity and acceleration of the randomized trajectory; (b) position, velocity and acceleration of the optimized trajectory.

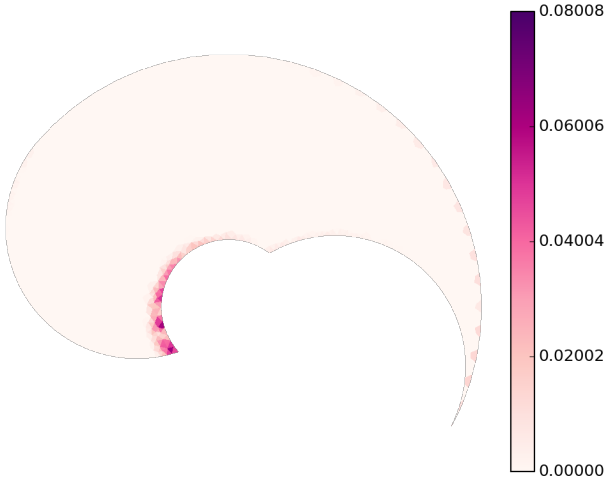


(a)

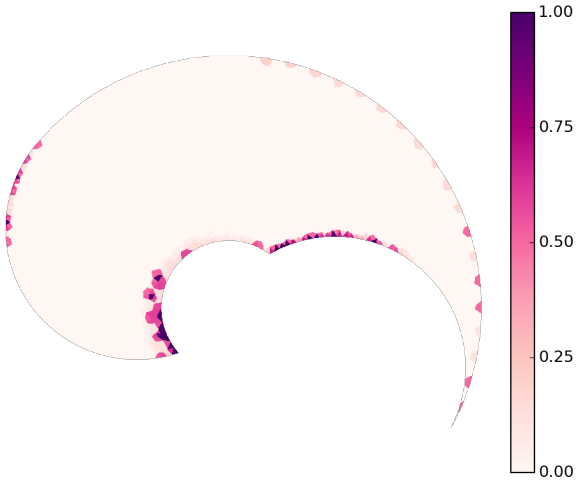


(b)

Figure C.11: Comparison between end effector kinematic profiles of the trajectories solving task 7 in Figure C.8. (a) end effector position of the randomized trajectory; (b) end effector position of the optimized trajectory.



(a)



(b)

Figure C.12: Comparison between interpolation error and trajectories obtained by means of the optimization and of the pseudoinverse / randomized approaches in the end effector space. (a) difference in the interpolation error between optimization and pseudoinverse / randomized approaches. Points which are more bright are characterized by the two approaches having the same interpolation error. The more darker the points, the higher the difference between the interpolation errors. (b) norm of the difference between trajectories generated by the optimization and the pseudoinverse / randomized approach. The darker the points, the more cumbersome the randomized trajectories. A cumbersome random trajectory will in fact generate a higher norm. The darkest points are those for which the pseudoinverse trajectory was not feasible and no randomized trajectory could be found. The norm is normalized between 0 and 1.

Point	err_{Iopt}	$err_{Ipi,r}$	$\|\|p_{opt}(t) - p_{pi,r}(t)\|\|$
34	7.09E-02	2.36E-13	16.26
35	6.79E-02	1.93E-13	17.52
36	6.48E-02	∞	∞
37	6.18E-02	∞	∞
38	5.87E-02	2.76E-13	19.12
39	5.57E-02	2.05E-13	18.18
40	5.26E-02	2.94E-13	21.05
456	8.01E-15	8.01E-15	0.00
556	1.19E-14	1.19E-14	0.00
626	1.66E-02	3.85E-13	14.55
884	9.89E-03	1.80E-13	16.19
968	2.57E-02	1.78E-13	22.79
969	2.06E-02	∞	∞
1175	3.27E-02	2.48E-13	17.59
1323	7.64E-15	7.64E-15	0.00
1528	2.98E-02	1.20E-13	9.29
1743	4.53E-02	2.73E-13	12.71
1785	4.45E-03	3.01E-13	17.65
2054	2.44E-04	3.78E-13	19.26
2101	1.49E-03	∞	∞
2190	2.61E-02	3.19E-13	19.71
2510	1.79E-02	∞	∞
2696	5.06E-02	∞	∞
2751	4.54E-02	1.42E-13	21.29
2774	6.70E-03	1.70E-13	23.01
2823	6.55E-15	6.55E-15	0.00
2841	3.83E-02	2.86E-13	20.27
2849	3.16E-02	1.81E-13	19.15
2891	3.29E-02	1.69E-13	21.33
2975	1.03E-14	1.03E-14	0.00

Table C.1: Interpolation error for the pseudoinverse / randomized and the optimization approaches and the norm of the difference between trajectories for 30 reaching tasks from the same point to destinations in the bottom-left region of Figure C.12, where the most difficulties for the pseudoinverse arise. err_{Iopt} is the interpolation error for the optimization approach, $err_{Ipi,r}$ the one for the pseudoinverse approach (or for the randomized one when the pseudoinverse trajectory was not feasible). $p_{opt}(t)$ is the position trajectory found by means of the optimization approach and $p_{pi,r}(t)$ the one found by means of the pseudoinverse or of the randomized approach.