UNIVERSITY OF GENOVA

PHD PROGRAM IN BIOENGINEERING AND ROBOTICS

Nonlinear dimensionality reduction for human movement analysis with application to body machine interfaces

by

Fabio Rizzoglio

Thesis submitted for the degree of *Doctor of Philosophy* (33° cycle)

February 2021

Prof. Maura Casadio	Supervisor
Prof. Ferdinando Mussa-Ivaldi	Supervisor
Prof. Giorgio Cannata	Head of the PhD program

Thesis Jury:

Prof. James Patton, *University of Illinois at Chicago*	External examiner
Prof. Silvestro Micera, *École Polytechnique Fédérale de Lausanne*	External examiner
Prof. Paolo Massobrio, *University of Genoa*	Internal examiner

Dibris

Department of Informatics, Bioengineering, Robotics and Systems Engineering

How we choose to feel
Is how we are

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements. This dissertation contains fewer than 65,000 words including appendices, bibliography, footnotes, tables and equations and has fewer than 150 figures.

Fabio Rizzoglio
February 2021

Table of contents

Abstract

For my PhD project, I set to explore how a nonlinear dimensionality reduction (DR) technique - autoencoder networks (AEs) - can identify low-dimensional latent manifolds of movement data. The thesis focuses on the application of this technique to body machine interfaces (BoMIs).

I begin by comparing nonlinear AEs to Principal Component Analysis (PCA), a linear DR method, in capturing essential information of kinematic signals, including hand gestures and object manipulations, as well as electromyographic signals (EMG) obtained from unconstrained movements of shoulders and arms. AEs exhibited higher performance than PCA in the reconstruction of hand kinematic and EMG data from a latent manifold. Therefore, a non-linear DR method has the potential to provide a more effective coding platform for human-machine interfaces (HMIs). I therefore investigated how the choice of hyperparameters (*e.g.*, type of activation function, number of hidden layers, etc.) affected the shape of the latent manifold, particularly its local curvature, and whether these potential effects were correlated with changes in reconstruction performance.

To gain more consistent insights on the structure of the nonlinear latent manifold, I developed a visual tool based on classic concepts of cartography. This display offers a direct and intuitive assessment of the AE's nonlinear transformation. I was able to demonstrate that the cartographic approach makes the visible structure of the latent manifold stable and independent of the AE's training parameters. As such, the proposed approach is a step toward defining a unique latent manifold.

After analyzing the properties of both linear (PCA) and nonlinear (AE) DR techniques, I focused on their applications within the control-scheme of a BoMI.

First, I aimed at providing BoMI users with the possibility to switch seamlessly between movement and EMG control. Such approach is essential to utilize the BoMI as a therapeutic tool for promoting recovery of muscle control after neurological injury.

In the clinical context it is essential to adapt the operation of the BoMI to the evolving state of its users. My guiding hypothesis for this purpose is that the operation of the interface is facilitated if the BoMI forward map is updated online to match the evolving latent manifold

of the user's motions. Results show that this adaptive approach increased the representational efficiency of the interface and significantly improved users' task-related performance.

As an extension of the application of AE-based BoMI, I developed a non-linear BoMI designed to control an assistive 4D virtual robotic manipulator and tested the interface on a cohort of unimpaired participants, who successfully acquired a high level of robot control.

To conclude, I investigated whether AEs can represent and estimate motor learning during the operation of a BoMI. This final study demonstrates that in fact this nonlinear method is effective to accurately track users' learning process.

Introduction

We live in a world where most of us can converse with virtual assistants, receive automatic song recommendations, and self-driving cars, once belonging in the pages of a science fiction novel, are a tangible part of our society. These incredible advancements in artificial intelligence (AI) technologies were made possible via the concurrent development of state-of-the-art machine learning (ML) techniques and the increased availability of large and complex data.

Most of the recent breakthroughs in AIs fall in the category of supervised learning, a paradigm where the AI system learns by examples and is specifically told by a human what to look for. For example, in order to teach a computer to recognize an image of a cat from that of a dog, thousands of labeled examples need to be manually crafted. Thus, one might argue that computers are not really intelligent, as a toddler can develop an understanding of what is a dog versus a cat after seeing only few pictures. Yann LeCun introduced a "cake analogy" to describe the current state of AI: "If intelligence is a cake, the icing on the cake is supervised learning, but the bulk of the cake is unsupervised learning".

Humans do not understand the world by sifting through massive, labeled data. Instead, we perceive the state of the world and leverage on our own reasoning to predict the consequences of our actions based on available information. Having an internal representation of the underlying structure of the world, we are able to resolve ambiguities in a sentence and infer missing information when provided with an incomplete text or an occluded image — a capability that machines lack. Today's AI technologies can easily classify images and recognize voices, but cannot understand Newton's second law or predict human movement. That is where unsupervised learning can fill the blanks. While algorithms based on supervised learning are taught to achieve an objective by human input, unsupervised ones extract patterns in data entirely on their own.

Nonetheless, the ability to extract such patterns, or even gain a simple conceptual understanding from complex data with a high number of features, becomes increasingly problematic when there are too many features (*curse of dimensionality*). However, some of

these features might not be as important as others. Sometimes the features themselves are correlated with each other. This is the case when we record the activity of a large number of neurons from the brain of humans, primates, or insects. Several studies have indeed proven that a much smaller number of dimensions than the number of the recorded neurons capture a large amount of variance in neural firing rates (Chapin and Nicolelis, 1999; Churchland et al., 2012; Matsumoto et al., 2005; Stopfer et al., 2003). These studies support the existence of a low-dimensional, *latent*, manifold that captures the covariance structure of the original signal (Cunningham and Byron, 2014; Gallego et al., 2017; Sadtler et al., 2014).

The problem of encoding the dynamics that underline high-dimensional processes within such latent manifold is a particular instance of unsupervised learning and is typically addressed by dimensionality reduction (DR) algorithms. DR techniques compress large sets of features onto a new feature subspace of lower dimension, the latent manifold, without losing the important information contained in the original signal, and, importantly, without requiring labels. As a result, DR can reduce the time that is required to train a supervised-ML model and can alleviate the problem of overfitting (Dosovitskiy et al., 2013). Most importantly, it improves the interpretability of the original input structure (Keim, 2002) and fosters learning of semantic feature representations of the input data. Among the numerous DR techniques available nowadays, there are linear methods, such as Principal Component Analysis (PCA) and Factor Analysis, as well as nonlinear ones, such as autoencoder (AE) networks (Kramer, 1991).

AEs are a class of artificial neural networks (ANNs) able to learn efficient representations of the data within their latent manifold without any supervision. AEs were introduced in the early 90's (Kramer, 1991; Sanger, 1989), right after the conception of the backpropagation algorithm (Le Cun, 1986; Rumelhart et al., 1985) as learning rule for updating the parameters of ANNs.

In recent years, together with the development in computational power and the creation of deep networks with more sophisticated structures (Agarap, 2018; Hinton and Salakhutdinov, 2006; LeCun et al., 2015) and optimization algorithms (Bottou, 2010; Kingma and Ba, 2014), AEs have become state-of-the-art for various applications, such as in image processing - for denoising (Vincent et al., 2010) and feature extraction (Masci et al., 2011; Vincent et al., 2008). In the field of neuroscience, AEs allow inferring latent dynamics from single-trial neural spiking (Pandarinath et al., 2018a,b) as well as extracting muscle synergies (Buongiorno et al., 2020). In addition, AEs can be used as generative models, due to their ability to randomly generate new data that look similar to the training data (Kingma and Welling, 2013; Makhzani et al., 2015). Hence, AEs can be used for adversarial domain

adaptation to improve the performance of speech recognition algorithms (Hsu et al., 2017) or to stabilize brain machine interfaces decoders over long periods of time (Farshchian et al., 2018).

A drawback of utilizing AEs is that their learning dynamics as well as their underlying mathematical structure are poorly understood (Fan et al., 2020; Vellido et al., 2012). In addition, the structure of non-linear AEs can rapidly get quite complicated when increasing the number of hidden layers/neurons. Deeper networks are more prone to get stuck in local minima (Hinton and Salakhutdinov, 2006), thus requiring further debugging time and state-of-the-art optimization techniques, whenever possible. Therefore, the use of linear methods might be preferred due to their ability of yielding more reliable manifolds than their non-linear counterparts (Patel et al., 2015; Yan et al., 2020).

In the light of these considerations, the question of whether one should choose the simplicity of linear models or the customizability and power offered by non-linear ones depends on the application context.

Narrowing it down to the field of human machine interfaces (HMIs), both methods have been successfully employed to build brain (BMIs) (Carmena et al., 2003) and body (BoMIs) (Casadio et al., 2012) machine interfaces. The approach followed by BMIs (Shenoy and Carmena, 2014) consists in decoding intended actions from high dimensional neural recordings, and then converting (or encoding) the decoded intentions into a lower dimensional set of commands for operating a device (*e.g.*, moving a computer cursor or robotic manipulator). Similarly, BoMIs extract a low dimensional control space from more down-stream information related to the execution of voluntary movements, such as body kinematics and muscle activity (Farshchiansadegh et al., 2014; Pierella et al., 2018). When designing a BMI/BoMI, the process of deriving the control signals from the latent manifold is, clearly, pivotal to ease the interaction between the user and the machine. On the one hand, with a linear control, users have the major advantage of being able to obtain a full repertoire of actions from the direct summation of simpler actions. On the other, linear models may overestimate the dimensionality of the motor primitives used to control the device, when such primitives lie on a nonlinear manifold (Tenenbaum et al., 2000), to the detriment of interface complexity as a larger signal space becomes necessary for control.

In my PhD project, I studied and compared the characteristic of linear and nonlinear DR by first investigating and comparing the properties underlying the learning mechanisms of each approach and then expanding the current perspectives of linear vs nonlinear controllers in various HMIs' contexts. In particular, I focused on the comparison between PCA and AEs. This thesis takes a bottom-up approach, starting from a basic question: do nonlinear AEs

outperform PCA when encoding information within signals derived from human movements? And if so, why? Then, it builds on these results and harnesses both PCA and AEs as computational engines of novel BoMIs. In particular, the increased customizability offered by AEs, with respect to PCA, is exploited to create (i) an adaptive interface based on the unsupervised tracking of latent manifolds and (ii) a platform to control an assistive device with multiple degrees of freedom (DoFs). Finally, my thesis investigates whether AEs can be used to represent motor learning and estimate user's behavior during the online operation of a BoMI.

Following a bottom-up approach, I divided the thesis into three parts:

- In **Part I**, I present the works that I conducted in order to describe the properties of PCA and nonlinear AE. Chapter 1 is dedicated to the comparison between nonlinear AEs and PCA in their ability to capture essential information of kinematic signals, including various hand gestures and object manipulations, as well as of EMG signals obtained from unconstrained movements of shoulders and arms. Chapter 2 expands on these concepts by proposing a tool to diagnose the mechanisms of learning of AEs via the visualization of their latent manifolds. In this chapter, I introduce a cartographic algorithm to obtain insights about the structure of the nonlinear latent manifold and about the information retained during the AE training. This provides a direct assessment of the extent to which the nonlinear analysis afforded by the AE outperforms the analysis provided by linear methods.

- In **Part II**, I focus on the application of PCA and AE within the control scheme of a BoMI. In Chapter 3, I present a novel linear BoMI, in which kinematic and muscular signals recorded from the upper body are combined into a hybrid signal to provide users with the possibility to seamlessly switch between movement and EMG control. In Chapter 4, I propose an original AE-based BoMI to control a computer cursor. Here, I also make use of AEs to design an adaptive interface for enhancing human-machine joint performance. In Chapter 5, I expand on the previous studies and use AEs to design a BoMI for controlling a virtual robotic manipulator with 4 DoFs.

- Finally, in **Part III**, I investigate whether AEs can be used to represent and estimate user motor learning. In Chapter 6, I present an approach that uses an AE to predict user's behavior and assess whether users developed a stable internal representation of the interface when operating via a non-linear BoMI.

List of figures

List of tables

Part I

Properties of linear and nonlinear dimensionality reduction

Preface

When dealing with high-dimensional signals, the existence of a low-dimensional latent manifold, capturing the relevant information of the original signal, has been demonstrated in several studies (Gallego et al., 2017; Gao and Ganguli, 2015; Sadtler et al., 2014; Stopfer et al., 2003). DR techniques are commonly used to encode the dynamics underlying high-dimensional processes within their latent manifolds (Cunningham and Byron, 2014), thus allowing to conveniently visualize them and to improve interpretability of the original input structure (Keim, 2002). Among the plethora of DR techniques available nowadays, autoencoder (AE) networks (Kramer, 1991) have seen a recent surge in popularity in the field of neuroscience (Farshchian et al., 2018; Pandarinath et al., 2018a,b). AEs are a class of artificial neural networks trained by unsupervised learning, reconstructing inputs in their outputs, without the requirement of labeled data. Most importantly, AEs can represent both linear and non-linear relations in the input data by making use of linear and non-linear activation functions. In the first part of the thesis, I focus on the application of AEs to extracting latent manifolds derived from body signals, such as kinematic data and muscular activities. My first goal was to evaluate whether a nonlinear AE was adequate to find a more parsimonious representation of such signals than a linear DR method, PCA. Chapter 1 is dedicated to address these points.

A drawback of utilizing AEs is that their learning dynamics as well as their underlying mathematical structure are poorly understood (Fan et al., 2020; Vellido et al., 2012). In other words, AEs can be seen as "black-boxes", as the internal mechanisms they use during training are shielded from human eyes, buried in layers of computations, making it hard to diagnose errors or biases. In this project, the goal of unveiling the hidden processes that take place during AE training was achieved via the visualization of the AE's latent manifold. In Chapter 2, I describe the diagnostic tool I designed for unveiling the black-box of non-linear AEs as well as for visualizing the resulting manifolds. I applied Multidimensional Scaling (MDS), in particular cartographic algorithms used for representing higher-dimensional maps in 2D, to visualize nonlinear latent manifolds derived from training AE on various synthetic

datasets. The goal was to capture the curvature of the latent manifold so that distances between points in the latent manifold reflect those between their decoded projections in the original signal space. The proposed approach provides us direct information about the effective non-linearity introduced by the dimensionality reduction algorithm. In addition, by looking at the latent signals at different stages of training, it is possible to obtain insights about the information that is retained during the AE training, thus highlighting the potential reasons why AEs outperform linear DR methods.

The training of an AE of a specific architecture with a given training set can result in different solutions Liao and Poggio (2017) due to differences in initial conditions, learning rate, stopping criteria, and other hyperparameters that are typically chosen so as to mostly minimize the reconstruction error (Claesen and De Moor, 2015). In Chapter 2, I further investigated how the choice of hyperparameters affected the shape of the latent manifold, particularly its local curvature, and whether these potential effects were correlated with changes in reconstruction performance.

Chapter 1

Linear and Non-linear Dimensionality Reduction Techniques on Full Hand Kinematics

1.1 Introduction

The complexity of the human hand makes it the subject of intensive research in prosthetics and robotics control. Controlling several degrees of freedom (DOFs)—there are 27 in each hand—can be a difficult task when both precision and speed are required as in dexterous prosthetic hand control. Since their first development in the 1940s, myoelectric prostheses, operated by electromyographic (EMG) signals, have undergone a series of design and control changes (Zuo and Olson, 2014).

Technological advances have resulted in more complex prostheses with an increased number of DOFs (Belter et al., 2013). The increase in design complexity was also associated with the high demand of prosthetic users to be able to perform dexterous tasks, such as handicrafts, operation of domestic and electronic devices, as well as dressing/undressing (Pylatiuk et al., 2007). For acceptable performance in such tasks, individual digit control is often required. Instead of allowing the independent control of each degree of freedom, currently available market options include a variety of prosthetic hands with a limited number

The content of this chapter has been published as Portnova-Fahreeva A., Rizzoglio, F., Nisky, I., Casadio, M., Mussa-Ivaldi, F.A. and Rombokas E., 2020. Linear and Non-linear Dimensionality-Reduction Techniques on Full Hand Kinematics. *Frontiers in Bioengineering and Biotechnology.*

of preset gestures associated with the most common grasp patterns to be performed in activities of daily living (ADLs). For example, the Michelangelo Hand (*Ottobock, Duderstadt, Germany*) includes seven grip patterns whereas its successor, the Bebionic Hand, from the same company includes 14 grip patterns.

While there have been many developments in the design of prosthetic hands, advances in control strategies have been limited. There are numerous types of controls used in upper-limb myoelectric prostheses from simple finite-state machines (FSM) to complex pattern recognition (PR) (Geethanjali, 2016). In FSM, usually two EMG signals are used to switch between grip patterns. This method can be effective for a small number of postures but in the case of 14 or even seven grips, this can be a strenuous and time-consuming task. In addition to slow controller speeds, prosthetic users have identified their myoelectric device speeds as inadequate for task completion (Pylatiuk et al., 2007). Such issues in the device performance could be contributing to the high abandonment rates that are prevalent in upper-limb prostheses (Biddiss and Chau, 2007).

Recently, a new type of prosthetic control has been proposed–*posture control* (Geethanjali, 2016). This capitalizes on the results of previous dimensionality-reduction studies performed on hand kinematics (Ingram et al., 2008; Santello et al., 1998; Todorov and Ghahramani, 2004). In these studies, Principal Component Analysis (PCA) was utilized to simplify the complex kinematics of hand grasps by finding a reduced number of linear combinations of input signals that explain most of the variability observed in grasping data. These combinations span the latent manifold of hand kinematics. By projecting the data along these directions, it is possible to obtain a compressed representation of hand configurations. A related approach to the control of prosthetic hands was introduced by Bicchi's group with the "Soft hand" (Della Santina et al., 2017, 2018). The underlying novel idea is that a variety of grasping patterns can be obtained by a single "synergy" of compliant actuators interacting mechanically with different shapes of grasped objects. While we recognize the value of this approach for the simplification of hand control during manipulation, here we are also considering the value of the hand as a communication device. In this broader scope, as well as in the performance with musical instruments, the ability to explicitly and precisely control finger configuration is essential. For this, we are considering in this study a data set obtained from American Sign Language (ASL).

In some myoelectric control methods, EMG signals are mapped to control parameters of the latent manifold, or principal components (PCs), which are derived through PCA (Matrone et al., 2012, 2010; Segil et al., 2013, 2015, 2016). An important difference of this method with respect to FSM and PR control is that posture control is continuous rather than discrete.

This allows users to directly operate the prosthetic device in a way more consistent with natural movement control instead of being limited to a finite set of pre-defined grasp options.

While PCA is a computationally straightforward and inexpensive procedure, it is limited by its ability to only account for linear relationships in the input signals. The assumption of linearity is not consistent with the geometry of hand kinematics.

To account for kinematic non-linearities, this study considers a dimensionality-reduction method based on autoencoder networks (AENs). AENs are artificial neural networks that are trained to reconstruct their inputs. They are composed of two parts: an encoder that converts the input data to a lower- dimensional, *latent*, manifold and a decoder that converts the latent manifold into the outputs. AENs provide an unsupervised method, reconstructing inputs in their outputs, without the requirement of labeled data. Most importantly, AENs are able to cope with both linear and non-linear relations in the input data by making use of linear and non-linear activation functions.

This study compares the performance of a non-linear AEN (nAEN) to that of PCA on examples of hand kinematics observed in human participants. In addition, it evaluates the case for the potential use of nAENs over PCA in a prosthetic controller.

1.2 Materials and Methods

1.2.1 Experimental Setup

Ten unimpaired right-handed individuals (six males, four females, 32.8 $\pm$ 9.4 years old) participated in this study. Participant recruitment and data collection conformed with the University of Washington's Institutional Review Board (IRB). Informed written consent was obtained from each participant. Basic measurements were taken from the right hand of each participant and recorded with other information.

The participants were first fitted with a right-handed data glove (*VirtualMotion Labs, Dallas, TX, USA*). A total of 20 signals were extracted from the glove that accounted for finger joint kinematics (Figure 1.1). The signals were recorded at a sampling rate of 100 Hz.

To calibrate the data glove, the participants were asked to perform a series of hand gestures presented to them on the screen. The movements were finger flexion/extension, finger abduction/adduction, thumb flexion/extension, and individual finger flexion against the thumb.

1 – thumb CMC_x
2 – thumb CMC_y
3 – thumb CMC_z
4 – thumb MCP_z
5 – thumb IP_z
6 – index MCP_z
7 – index PIP_z
8 – index DIP_z
9 – abduction/adduction angle between index and middle
10 – middle MCP_z
11 – middle PIP_z
12 – middle DIP_z
13 – ring MCP_z
14 – ring PIP_z
15 – ring DIP_z
16 – abduction/adduction angle between middle and ring
17 – pinky MCP_z
18 – pinky PIP_z
19 – pinky DIP_z
20 – abduction/adduction angle between ring and pinky

Notes:
- IP – interphalangeal
- CMC – carpometacarpal
- MCP – metacarpophalangeal
- PIP – proximal interphalangeal
- DIP – distal interphalangeal
- z-rotation represents flexion/extension of a joint
- Thumb CMC joint includes signals about x-, y-, z-rotations

Figure 1.1 Virtual Motion Labs Glove used for the study to record kinematics of the right hand. Numbers on the glove represent the kinematic signals that were extracted and recorded for analysis.

Once the glove was calibrated, the participants were guided through a sequence of hand movements consisting of: (i) ASL Gestures, (ii) Object Grasps, and (iii) ADL Tasks (Figure 1.2) as described in the following.

ASL Gestures

During the first phase of the experiment, the participants were asked to perform 10 different ASL gestures (Figure 1.2A). They were asked to repeat each gesture 10 times. Each gesture repetition counted as a trial. At the beginning of each trial, the participants started in the REST position with their elbows on the table and the right hand raised straight up. When presented with a gesture, the participants were given 3s to mimic it as displayed on the screen and maintain the gesture until instructed to return to REST position. After 3s in the REST position, a new trial would start. The participants had an opportunity to practice the gestures before the beginning of the phase.

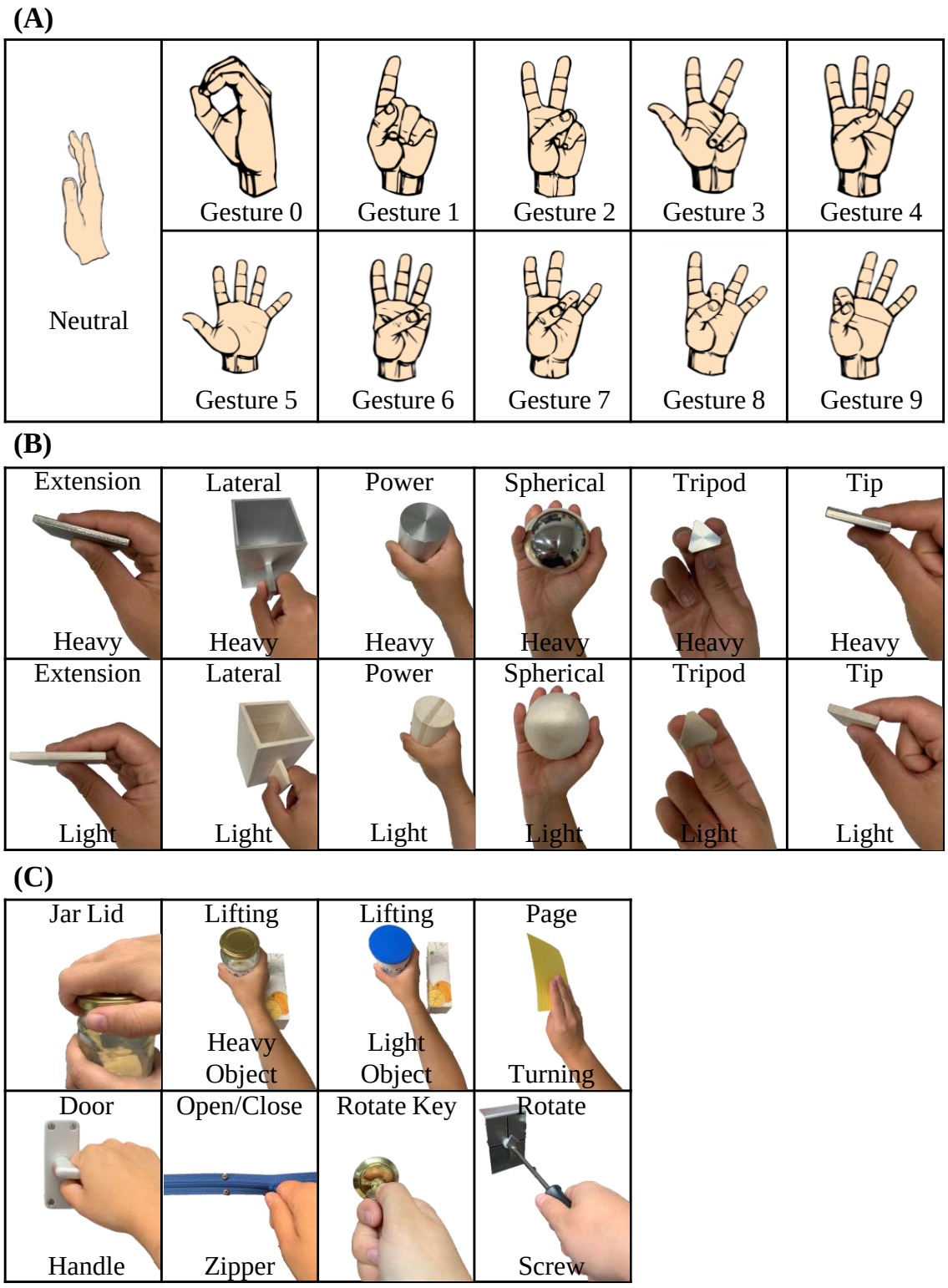

Figure 1.2 Study setup consisting of three different phases: (A) American Sign Language (ASL) Gestures; (B) Object Grasps; (C) Activities of Daily Living (ADL) Tasks.

Object Grasps

During the second phase of the experiment, the participants were asked to perform a series of object grasps from the Southampton Hand Assessment Procedure (SHAP) (Figure 1.2B) (Light et al., 2002). The testing board was placed 8 cm from the edge of the table closest to the participant. The board was then aligned so that the target object was directly in front of the participant. There were 12 objects to be grasped with six different grasping types. Furthermore, each object could be either light or heavy.

Each object had to be grasped 10 times. Before grasping a new object, video instructions were shown to the participants on the required way of grasping. They could then practice grasping under the supervision of the experimenter to ensure a correct and consistent execution.

The participants were given 5 s to complete each grasp starting and ending the grasp on the REST position. In the REST position, both participant's hands lied prone on the table. Between each trial, there was a 5 s resting period.

ADL Tasks

During the third phase of the experiment, the participants performed the ADL portion of the SHAP. A total of eight different tasks were selected for this phase (Figure 1.2C). Each task was performed 10 times. The participants were given 7 s to perform each task with a 5 s of REST time between each trial.

As in the second phase, the participants were shown a video with instructions on how to appropriately complete the task. They were then instructed to practice the task until ready. The testing board was placed 8 cm away from the edge of the table closest to the participant. During REST, the participants held the hands supine on the table to the sides of the testing board.

1.2.2 Data Processing

For each phase, data were recorded during both REST (when participants were instructed to be in REST position) and ACTIVE (when participants were instructed to perform the given task) conditions. Only ACTIVE conditions were used for data analysis.

Preprocessing

The recorded data were filtered with a first-order Butterworth filter in MATLAB (*MathWorks, Natick, MA, USA*). The cutoff frequency was 10Hz. REST data were removed from analysis, and the remaining data points were labeled to indicate different hand movements and trial numbers for each participant. Trials, in which participants did not complete the movement as requested, possibly due to loss of attention or inability to understand the given task in due time, were excluded from the analysis.

The data were then arranged into four datasets: *ASL Gestures, Object Grasps, ADL Tasks*, and *Combined*. The Combined dataset contained data from *ASL Gestures, Object Grasps*, and *ADL Tasks* for each participant.

Each signal was normalized by the absolute maximum value across all signals in each dataset prior to analysis (Sola and Sevilla, 1997). Data of each participant were randomly split into training (80%) and testing (20%) using a holdout method (Oxford and Daniel, 2001). Training samples were used to generate a model, PCA or nAEN.

1.2.3 Data Analysis

To study the effects of linear and non-linear dimensionality- reduction methods, we compared the performance of PCA and nAEN algorithms with two, three, four, five, and six latent dimensions [PCs and coding units (CUs), respectively].

PCA was performed using the built-in MATLAB function pca, which is based on singular value decomposition (Wall et al., 2003). Dimensionality reduction using nAEN was performed using TensorFlow, a Python (*Python Software Foundation, DE, USA*) library for machine learning applications developed by Google Brain (Abadi et al., 2016).

nAEN Architecture

The basic nAEN structure used for this experiment included a total of three hidden layers, the middle one being the bottleneck layer (Figure 1.3). Similarly to the original AEN proposed by (Kramer, 1991), we chose a non-linear activation function for the first and third hidden layers, and a linear activation function for the bottleneck. The use of both linear and non-linear activation functions had been shown to increase the ease with which the network learns linear relationships in the data (Haesloop and Holt, 1990). The transformations from the normalized input X to the output data $\hat{X}$ through the encoder (Equations 1.1, 1.2) and decoder (Equations 1.3, 1.4) parts of nAEN are:

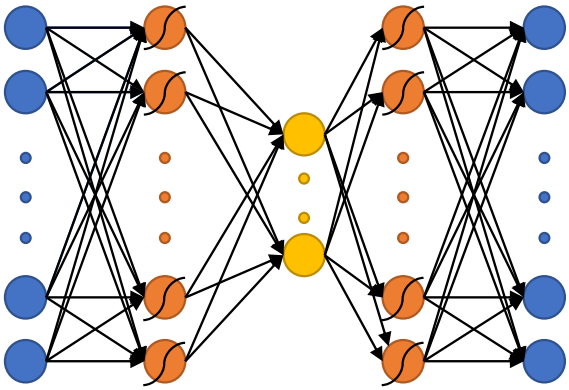

Figure 1.3 The non-linear Autoencoder Network (nAEN) structure used in this study. Curved lines over neurons represent that non-linear activation functions (*i.e.*, hyperbolic tangent) were used to calculate that layer. Otherwise, activation functions were linear.

$$layer_1 = tanh(X * w_1 + b_1) \tag{1.1}$$

$$layer_2 = layer_1 * w_2 + b_2 \tag{1.2}$$

$$layer_3 = tanh(layer_2 * w_3 + b_3) \tag{1.3}$$

$$\hat{X} = layer_3 * w_4 + b_4 \tag{1.4}$$

where w_i were the weights and b_i were the biases found during network modeling. While the main purpose of the study was to compare the performance of linear and non-linear dimensionality-reduction algorithms, it is important to note that neural network structures, such as AENs, can be optimized further for improved performance. A simple nAEN structure was chosen for this study while more complex structures can be explored. Neural networks can have multiple layers, various structures, and many hyperparameters that directly affect the performance of the network.

Full-batch gradient descent was used for training the network.

Tuning of hyperparameters, such as the learning rate, number of steps, type of non-linear activation functions, and regularization, was performed (Figure 1.4). A separate validation dataset from a participant, whose data were not used in the experiment for overall analysis, was utilized for hyperparameter tuning. In this dataset, the participant (P0) performed all the tasks of the ASL Gestures phase, 10 trials each. A 5-fold cross- validation (CV) was

conducted (Oxford and Daniel, 2001), and the performance of each hyperparameter pair was evaluated using Variance Accounted For (VAF) (see section Performance Metrics for more information on VAF).

Learning rates of 0.01 and 0.025 produced the most stable results across the variety of number of steps tested (Figure 1.4A). The largest learning rate of 0.05 produced the worst results while the smallest learning rate of 0.001 produced the best results for the number of steps over 10,000. The nAEN performance difference was minimal across the learning rates of 0.01, 0.05, and 0.001 for 10,000 steps. Although 20,000 steps resulted in a slightly improved performance of the network for the learning rate of 0.001, a larger number of steps would lead to longer training times for nAEN. Since it was of interest to optimize both the network performance as well as the training times, the learning rate of 0.01 and 10,000 steps were chosen for this study.

Sigmoid (*sig*), hyperbolic tangent (*tanh*), and rectified linear unit (*relu*) performed similarly for the learning rate of 0.01 (Figure 1.4B). As a result, we were free to choose any non-linear activation function for the nAEN structure, and the hyperbolic tangent was chosen for this experiment.

Adaptive Moment Estimation (ADAM) optimizer was used to speed up the training of the nAEN (Kingma and Ba, 2014). No regularization was used in training the model for this study, as we did not find any evidence of overfitting for our dataset (Figure 1.4C).

For nAEN, the weights and biases for all models created for each participant were initialized in the same way for comparative purposes.

1.2.4 Performance Metrics

The performance in terms of VAF was evaluated on the testing samples. Variability of dimensions as well as visualization and separability of movements in the latent and reconstructed manifolds were tested on the entire dataset (training and testing combined).

Dimensionality Reduction

VAF measures the difference between reconstructed output and original input signals. It was chosen to capture the ability of a dimensionality-reduction method to reconstruct the desired signal from the latent manifold. VAF offers a measure of the information preserved by the dimensionality-reduction algorithm and is directly related to reconstruction error (Equation 1.5). A VAF of 100% indicated that the output and the input were identical.

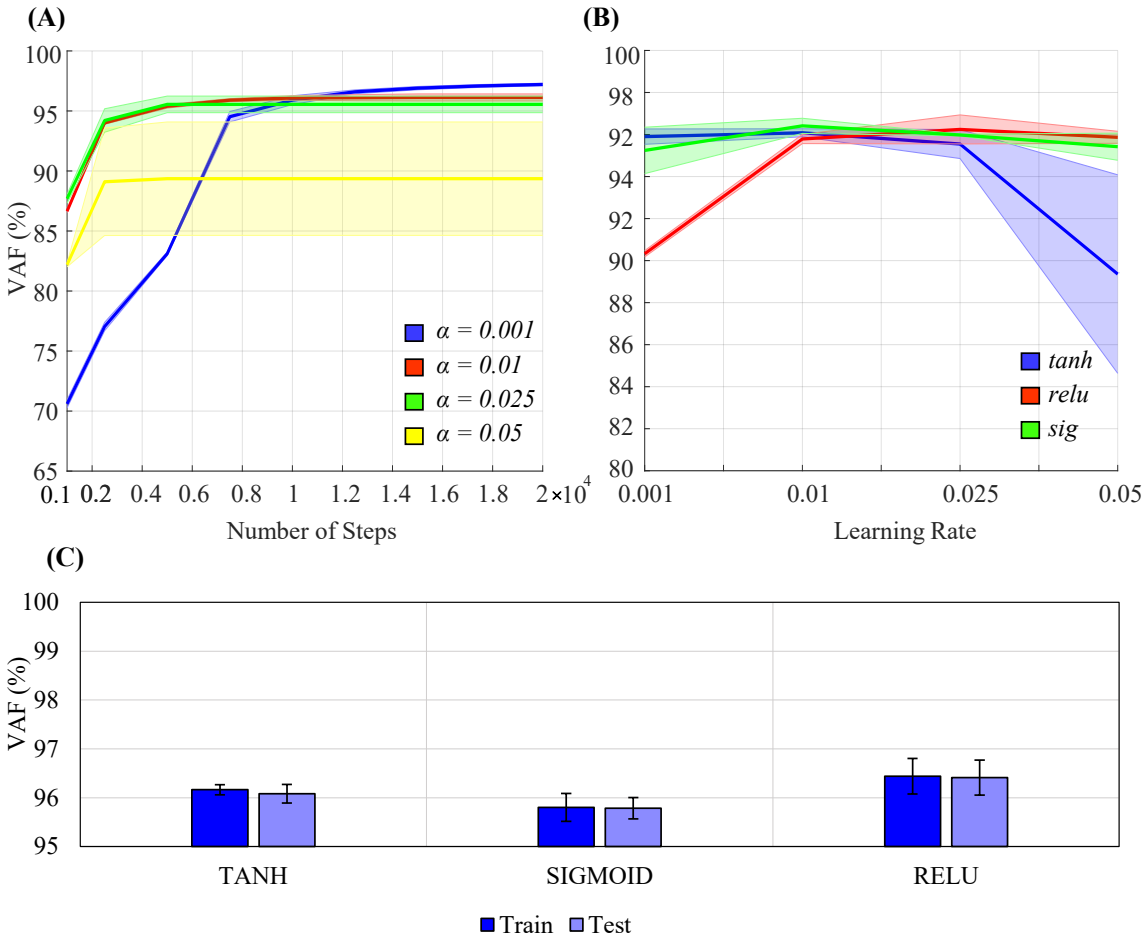

Figure 1.4 Effects of certain hyperparameters on the non-linear Autoencoder Network (nAEN) performance: (A) number of steps and learning rate (0.001, 0.01, 0.025, 0.05); (B) learning rate and type of non-linear activation function (tanh, relu, sig). The performance is calculated using Variance Accounted For (VAF), which represents the difference between the output and the input. Solid lines represent average results across all participants. Faint shaded lines represent 95% confidence interval. (C) nAEN performance difference on training and testing datasest for different activation functions with no regularization. Error bars represent 95% confidence intervals. The nAEN performance difference was minimal across the learning rates tested for over 10,000 steps. The difference in the performance was minimal for the learning rate of 0.01 across the non-linear activation functions tested. No evidence of overfitting of the dataset was found.

$$VAF(\%) = \left(1 - \frac{var(X - \hat{X})}{var(X)}\right) * 100 \tag{1.5}$$
$$X - original\,data$$
$$\hat{X} - reconstructed\,data$$

In addition, data variance across three datasets (ASL Gestures, Object Grasps, ADL Tasks) was calculated for each participant. It was done by first calculating the variance of each of the 20 kinematic signals across all samples in the input data. The variance values were then averaged across 20 signals to produce one value of variance for each participant. The correlation between data variance and resulted VAF with two latent dimensions was calculated for both PCA and nAEN.

Lastly, to explore the reconstructing performance of nAEN and PCA, the second gesture of the ASL Gesture dataset ("gesture signifying number one") was reconstructed from two-dimensional latent manifolds of these dimensionality-reduction methods. To visualize the reconstructed gesture, one of the Leap Motion (*Leap Motion Inc., San Francisco, CA, USA*) hand models was utilized in Unity (*Unity Technologies, San Francisco, CA, USA*). A few snapshots were taken as the reconstructed gesture went from the REST into the ACTIVE positions.

Dimension Variance

Dimension variance was the variance associated with each dimension in the latent manifold of PCA and nAEN. To calculate dimension variance, the input data were reduced to PCs and CUs for each participant across each of the four datasets. For PCA, it was done in the following way, where *eig* produces a diagonal matrix D of eigenvalues of a covariance matrix, cov, of normalized data X (Equation 1.6).

$$D = eig(cov(X)) \tag{1.6}$$

Each PCA dimension, PC_i, where i was the dimension number, was found by sorting the diagonal matrix D in the descending order and taking the i^{th} column of the sorted matrix (Equation 1.7).

$$PC_i = diag(D(:,i)) \tag{1.7}$$

For nAEN, latent dimensions were calculated by passing the normalized data X through the *encoder* part of the network (Equations 1.1, 1.2). CUs were the corresponding columns of $layer_2$ (Equation 1.8).

$$CU_i = layer_2(:, i) \tag{1.8}$$

Each latent dimension (PCs or CUs) was represented by A_i, an mxn matrix, where m was the number of observations and n was the number of latent dimensions. The mean of each latent dimension was calculated (Equation 1.9).

$$\mu = \frac{1}{m}\sum_{j=1}^{m} A_j \tag{1.9}$$

Afterwards, the variance of each latent dimension, v_i, was calculated (Equation 1.10).

$$v_i = \frac{1}{m-1}\sum_{j=1}^{m} \|A_j - \mu\|^2 \tag{1.10}$$

Lastly, what was defined as dimension variance in this study, v_{dim}, was calculated by determining the percentage of v_i with respect to the overall variance of all considered latent dimensions (Equation 1.11).

$$v_{dim} = \frac{v_i}{\sum_{i=1}^{n} v_i} \tag{1.11}$$

Dimension variance was ordered in descending order for visualization and comparative purposes.

In addition, the average difference between dimensions with the highest and lowest values of variance ($range_{dim,avg}$) was calculated across all 10 participants for each dataset and called *range of dimension variance* (Equation 1.12).

$$range_{dim,avg} = \frac{1}{n}\sum_{dim=1}^{n} max(v_{dim}) - min(v_{dim}) \tag{1.12}$$

Latent Trajectories Visualization

To aid the visualization of the latent trajectories, we focused on manifolds with two dimensions. To visualize the PCA latent trajectories, the input data was first reduced to 2 PCs. The PC pairs for each dataset and each participant were then plotted on a 2D surface where PC1 represented the x-axis and PC2 represented the y-axis. Each sample representing a kinematic instance in 20D space was plotted as a point in this 2D graph.

A similar technique was utilized for visualizing the latent trajectories of nAEN. After reducing the input data to the latent manifold, the pair of CUs for each dataset was plotted on a 2D linear surface where CU1 represented the x-axis and CU2 represented the y-axis.

Separate movements for each dataset were plotted with a different color for ease of differentiation.

Movement Separability

Movement separability was defined as a measure of distance between movements (from the 20D input, latent manifold, or reconstructed 20D data) that allowed a simple classification algorithm to differentiate between given classes of postures. Movement separability was calculated for all datasets, except for Combined, using SoftMax regression (Gao and Pavel, 2017). SoftMax regression was chosen as a simple example of a linear classification algorithm that did not require hyperparameter tuning such as Dense Neural Networks (DNNs), which can also be used for classification purposes (Schmidhuber, 2015). The aim was to develop a simple understanding of the linear separability of different movements across latent manifolds of nAEN and PCA. Higher accuracy percentage indicated a latent manifold in which classes (*i.e.*, hand movements) were more linearly separable.

When designing a controller, creating a space where different movements can be easily separated can be of high importance. When navigating along a more separable control space, the user might have the ability to switch between different tasks and/or movements much faster than in cases where tasks are less separable.

SoftMax regression was applied to the 2CUs/PCs latent manifolds, as well as to the reconstructed 20D data for nAEN and PCA and the original input data for each participant and dataset. Assessing the separability of reconstructed space might be important in understanding how data variability is preserved upon reconstruction in both nAEN and PCA. We used a 5-fold CV to calculate the accuracy on each dataset.

1.2.5 Statistical Analysis

We used MATLAB Statistics Toolbox functions and custom- written code for our statistical analysis. The normality was tested by applying the Anderson-Darling (AD) Test (Anderson and Darling, 1954). When the normality assumption was violated, we used rank statistics for our statistical analysis. This was the case for the VAF, range of dimension variance, and classification accuracy.

We used the Wilcoxon Signed-Rank Test (Wilcoxon, 1992) to understand the differences in VAF between the dimensionality-reduction methods, and the Friedman's Test (Friedman, 1937) to understand the effect of datasets (*e.g.*, ADL Tasks, ASL Gestures, etc.) on VAF. In a post-hoc analysis, Bonferroni correction (Bonferroni, 1936) was used to verify statistically significant differences among datasets within nAEN and PCA.

We also used the Wilcoxon Signed-Rank Test to compare the differences in range of dimension variance between two dimensionality-reduction methods across all latent dimensions and datasets, and for the comparison of SoftMax classification accuracy between reduced-dimension (2D) and reconstructed (20D) data for both nAEN and PCA.

In all our analyses, the level of significance was set to 0.05. After the Bonferroni adjustment, the level of significance for the post-hoc analysis was set to 0.0125 (0.05/4)

1.3 Results

1.3.1 Dimensionality Reduction

For all four datasets, nAEN outperformed PCA by reconstructing the input data with higher VAF for two, three, four, five, and six latent dimensions (Figure 1.5). The average VAF with just 2CUs across all datasets was 94% for nAEN whereas it was 78% with 2PCs for PCA.

Wilcoxon Signed-Rank Test revealed an effect on VAF by the dimensionality-reduction method ($p < 0.001$). When comparing nAEN and PCA, the difference in the performance between the two methods decreased as the number of dimensions in the latent manifold increased (Figure 1.5).

By performing the Friedman's Test on VAF produced by PCA and nAEN models with two-dimensional latent manifolds, an effect of datasets was revealed on both dimensionality-reduction methods ($p < 0.001$). Specifically, PCA performed similarly for both Object Grasps and ADL Tasks datasets ($p = 0.986$), but its performance decreased significantly for ASL Gestures ($p = 0.002$ and $p = 0.006$ when compared to Object Grasps and ADL Tasks, respectively). The nAEN, on the contrary, had higher performance for ASL Gestures over two dimensions in comparison to ADL Tasks ($p < 0.001$) and Combined ($p < 0.001$) datasets.

Accordingly, the correlation between VAF and signal variance of each dataset was analyzed (Figure 1.6A). The R^2 values for a linear model between two variables were 0.58 and 0.54 for nAEN and PCA, respectively. The relationship is positive for nAEN and negative for PCA (observed from the slopes of the linear regression lines). Lastly, from Figure 1.7,

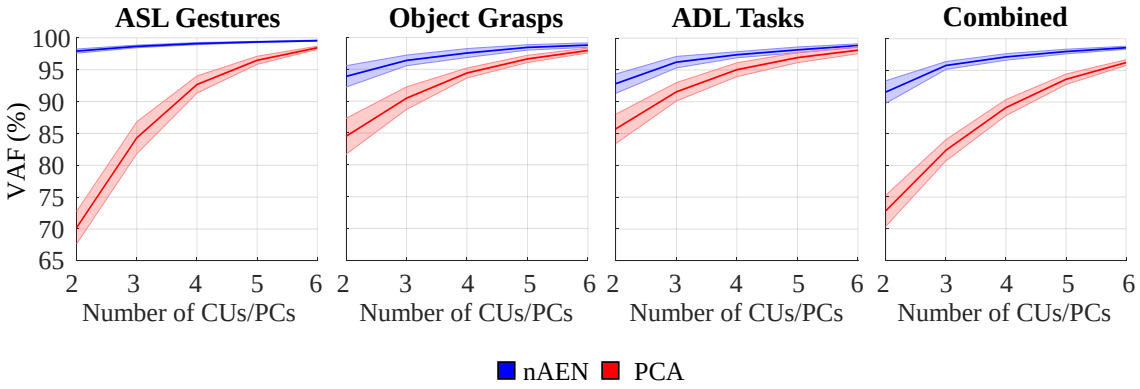

Figure 1.5 Performance of non-linear Autoencoder Network (nAEN, blue) vs. Principal-Component Analysis (PCA, red) with 2, 3, 4, 5, and 6 dimensions in the bottleneck layer on four different datasets: American Sign Language (ASL) Gestures, Object Grasps, Activities of Daily Living (ADL) Tasks, and Combined. The performance is calculated using Variance Accounted For (VAF), which represents the difference between the output and the input. Solid lines represent average results across all participants. Faint shaded lines represent 95% confidence interval. nAEN outperformed PCA across all datasets and for all number of latent dimensions tested in this study.

one can observe the visual differences in the performance of two dimensionality-reduction methods. While nAEN (middle column) was able to closely match the original hand gesture (left column), PCA (right column) failed to reconstruct the proper flexion of middle, ring, and pinky fingers as well as the full extension of the index finger. Reconstruction of all gestures and trials is shown in Supplementary Video.

1.3.2 Dimension Variance

Variance spread across nAEN and PCA dimensions was plotted for participant P1 performing ASL Gestures (Figure 1.8). Variance dropped significantly for each new added PC dimension whereas nAEN exhibited a less prominent decrease in variance for every additional latent dimension. The sharper drop in PCA variance appears to be a consequence of the orthogonality of subsequent eigenvectors.

Such behavior was consistent across all participants, as described by the average difference between dimensions with the highest and lowest variance (Figure 1.9). This difference was greater for PCA than for nAEN and significant in all conditions ($p = 0.002$), with the exception of ASL Gestures reconstructed with 2CUs/PCs ($p = 0.106$), according to Wilcoxon Signed-Rank Test.

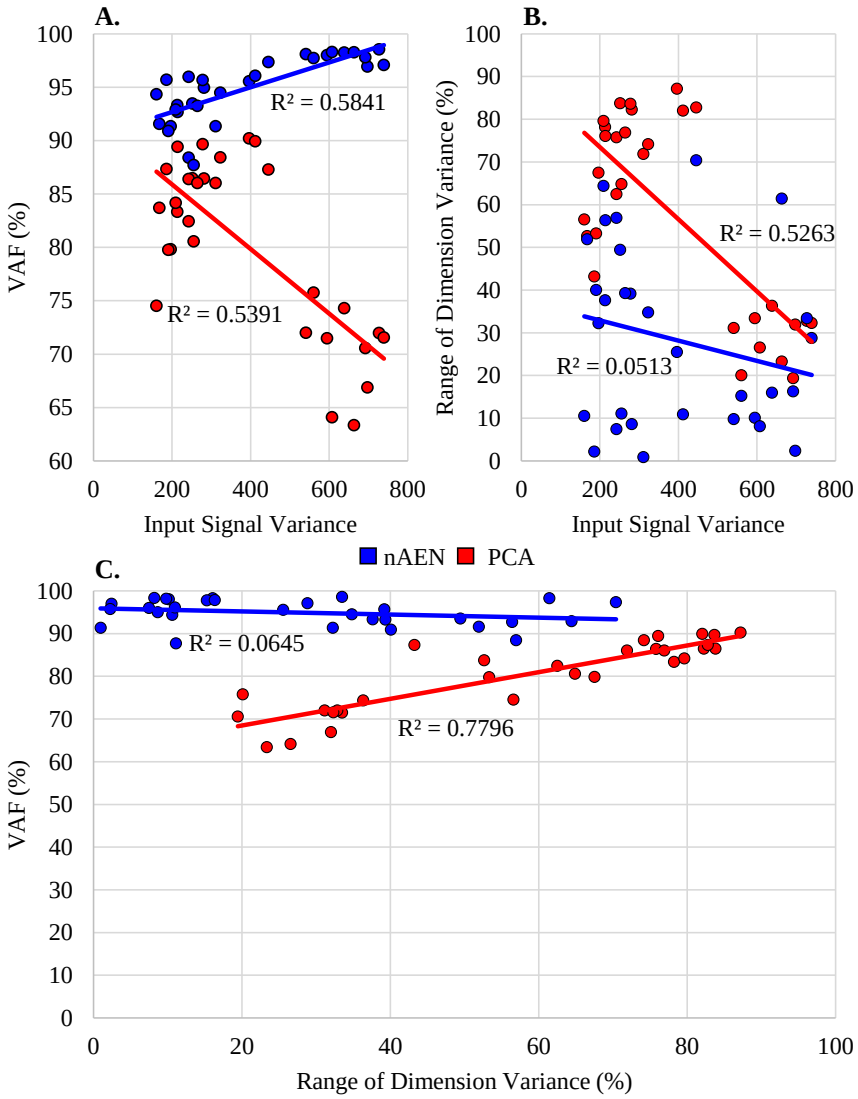

Figure 1.6 Correlation between average signal variance of input data across 10 participants and Variance Accounted For (VAF) with two-dimensional latent manifold plotted for three different datasets (American Sign Language (ASL) Gestures, Object Grasps, Activities of Daily Living (ADL) Tasks) for non-linear Autoencoder Network (nAEN, blue) and Principal Component Analysis (PCA, red). (B) Correlation between average signal variance of input data across 10 participants and range of dimension variance for 2 Coding Units (CUs)/Principal Components (PCs). (C) Correlation between VAF and range of dimension variance for 2CUs/PCs for three datasets (ASL Gestures, Object Grasps, ADL Tasks). PCA exhibited some correlation across all three scenarios tested whereas nAEN experienced no correlation.

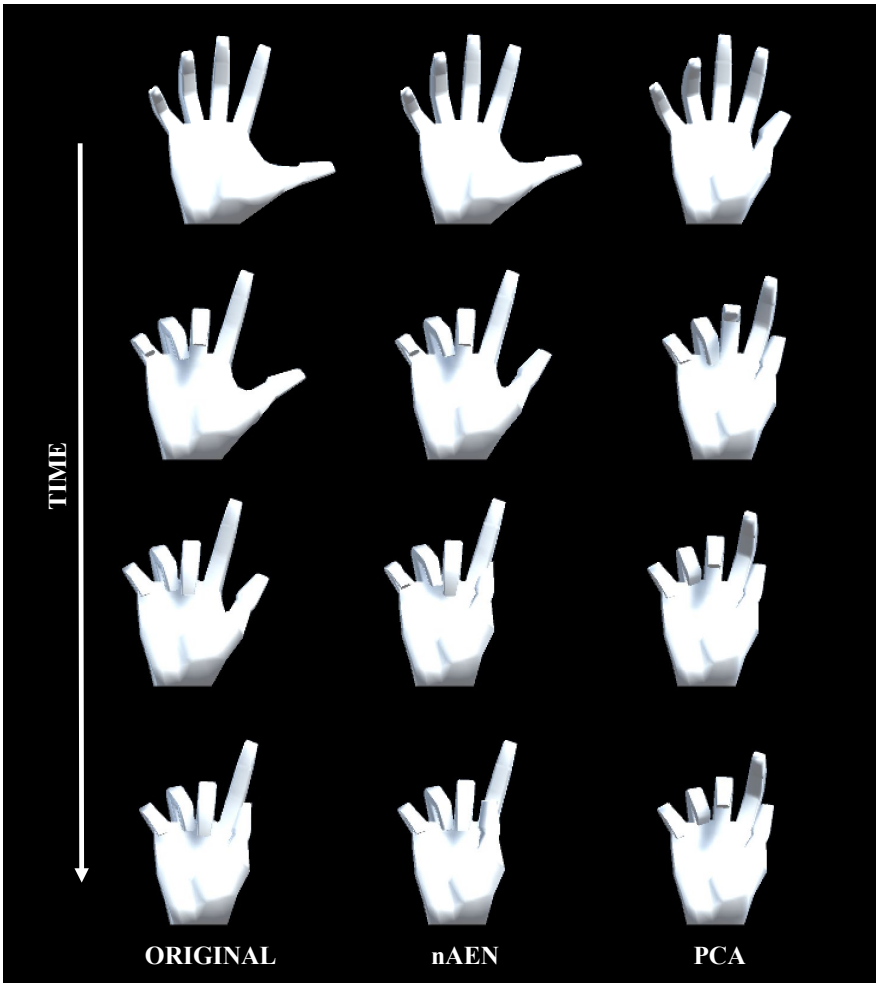

Figure 1.7 Visualization of gesture 1 from American Sign Language Gestures dataset, reconstructed from two-dimensional latent manifolds of non-linear Autoencoder Network (nAEN, middle column) and Principal Component Analysis (PCA, right column). The reconstructed gestures were compared to original gesture (left column). A few snapshots of each gesture were taken in time from REST to ACTIVE states.

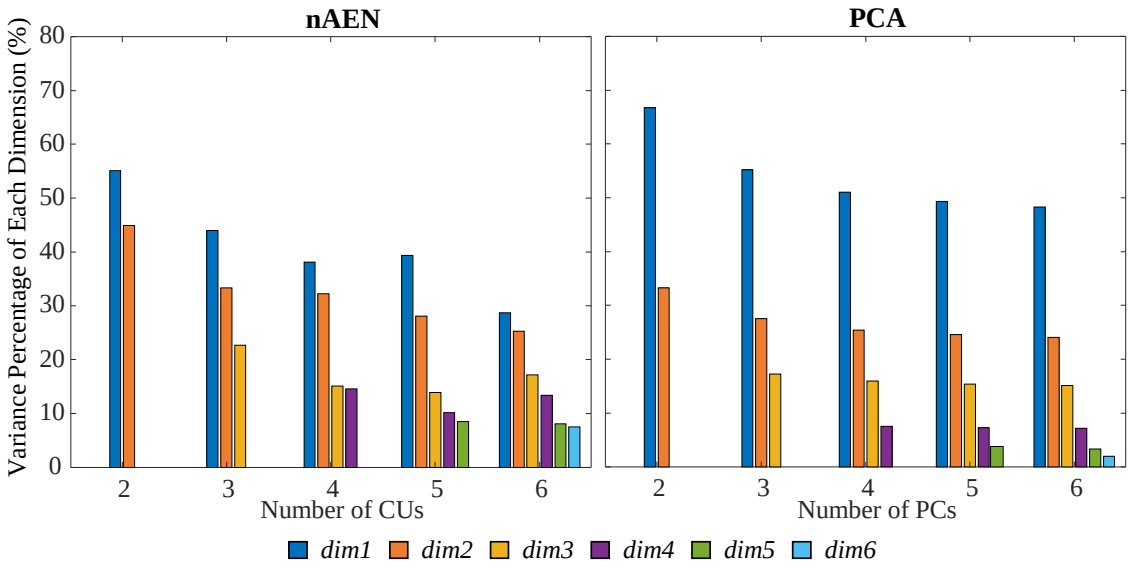

Figure 1.8 Variance percentage of each latent dimension (represented with different colors) with respect to the overall variance of the dimensions considered in the analysis [number of Principal Components (PCs)/Coding Units (CUs)] for P1 performing American Sign Language (ASL) Gestures. Two, three, four, five, and six dimensions in the bottleneck are compared for non-linear Autoencoder Network (nAEN) and Principal Component Analysis (PCA) and ranked by order of decreasing variance. nAEN exhibited smaller drop of variance across latent dimensions in comparison to PCA for P1.

In addition, an interesting observation could be made in regard to the average difference between latent dimensions with highest and lowest variance across three datasets (ASL Gestures, Object Grasps, ADL Tasks). When plotting the range of dimension variance vs. the signal variance across each dataset, a correlation is revealed for the PCA case–R^2 of 0.53 (Figure 1.6B). No such correlation was found for nAEN (R^2 of 0.05). The relationship found for PCA was inverse: higher signal variance dataset resulted in lower difference between latent dimensions with highest and lowest variance.

Lastly, plotting the VAF of each participant for the three datasets vs. the range of dimension variance for 2CUs/PCs reveals similar correlation for PCA ($^R2 = 0.78$) and no correlation for nAEN ($R^2 = 0.06$) (Figure 1.6C).

1.3.3 Latent Trajectories Visualization

nAEN and PCA latent trajectories were visualized for 2CUs/PCs in the case of participant P1 performing ASL Gestures (Figure 1.10A), Object Grasps (Figure 1.10B), and ADL Tasks

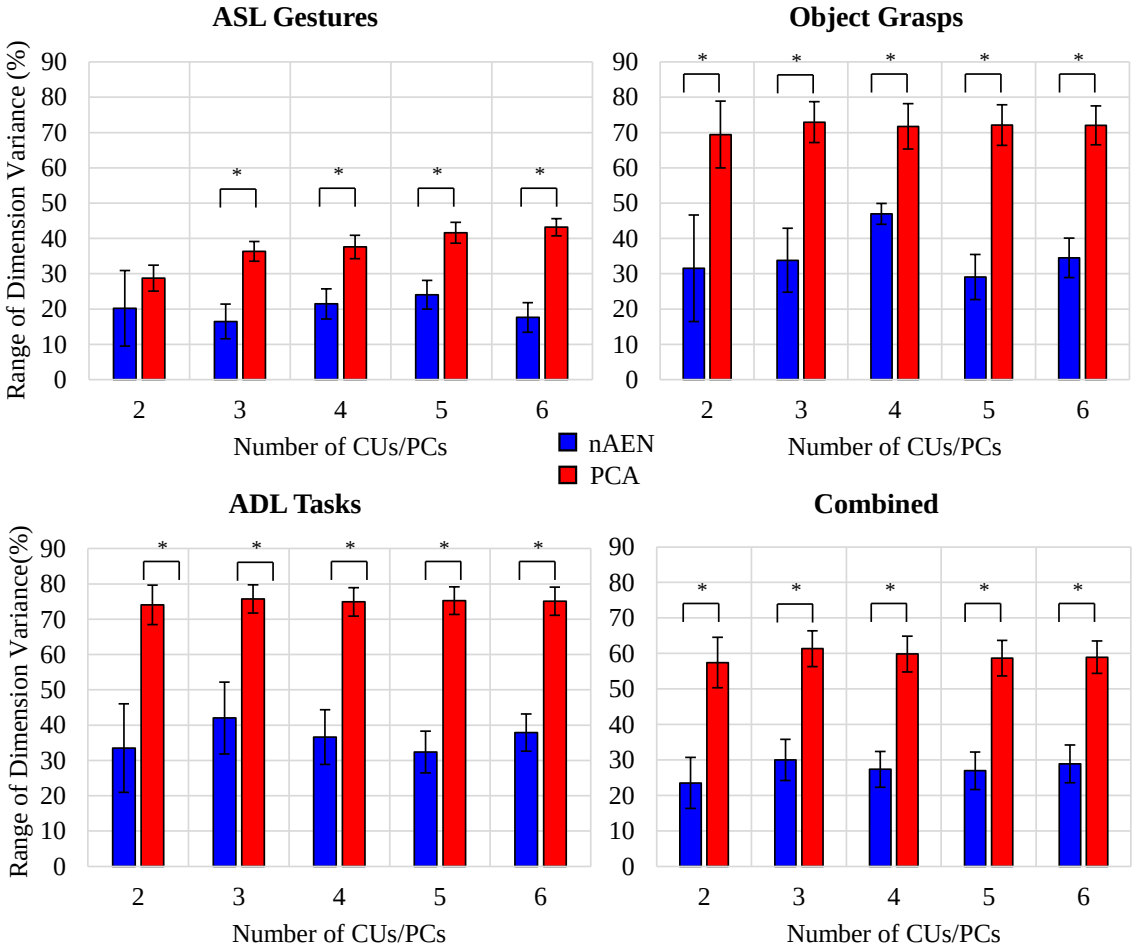

Figure 1.9 Difference between dimensions with highest and lowest variance values averaged across all participants for four datasets: American Sign Language (ASL) Gestures, Object Grasps, Activities of Daily Living (ADL) Tasks, and Combined. Statistical significance: $p = 0.106$ for ASL Gestures for 2CUs/PCs; *indicates statistical significance of $p = 0.002$. nAEN exhibited smaller drop of variance across latent dimensions in comparison to PCA across all datasets and for all number of latent dimensions except for 2 dimensions in ASL Gestures.

(Figure 1.10C). All trials used in the analysis were plotted, and separate gestures were indicated using different colors.

The overall structures of the plotted latent trajectories were similar across PCA and nAEN, with different gestures visually separated. Some movements appeared closer to each other in the 2D manifold than to other movements. Certain ASL gestures (*e.g.*, gestures 2 and 3) appeared closer to each other than to others in both nAEN and PCA manifolds. In Object Grasps, heavy and light versions of the same grasp took the same part of the visualization space. ADL tasks that required similar type of grasping (*e.g.*, using a screwdriver and opening a jar lid) appeared in the same part of the 2D space.

1.3.4 Movement Separability

SoftMax regression on the 2D manifold of nAEN and PCA did not reveal any significant difference in separability of movements between the two methods (Figure 1.11). The difference was insignificant across all three datasets ($p = 0.846$ for ASL Gestures, $p = 0.695$ for Object Grasps, and $p = 0.557$ for ADL Tasks).

There was a significant difference ($p = 0.002$) in the classification accuracy when the 20D manifolds, reconstructed from two latent dimensions, were evaluated. Across all three datasets, nAEN generated more separable representations than PCA. There was no significant difference between the separability of the reconstructed 20D and reduced 2D representations with PCA across ASL Gestures ($p = 0.695$), Object Grasps ($p = 0.492$), and ADL Tasks ($p = 0.695$).

Lastly, when applying SoftMax regression on the original input 20D kinematic space, the separability of classes was higher than that of the reconstructed 20D and the reduced 2D manifolds for both PCA and nAEN. The classification accuracy was at nearly 100% across all datasets (Figure 1.11).

The overall separability was much higher for ASL Gestures than for Object Grasps or ADL Tasks datasets for the reduced 2D and reconstructed 20D spaces (Figure 1.11).

1.4 Discussion

The complexity of a human hand makes the control of its prosthetic analog a challenging task. While posture control has been shown to be a novel and innovative way of providing a continuous controller for prosthetic users of their highly sophisticated devices, it has been limited by the linear nature of its underlying dimensionality-reduction algorithm. In this

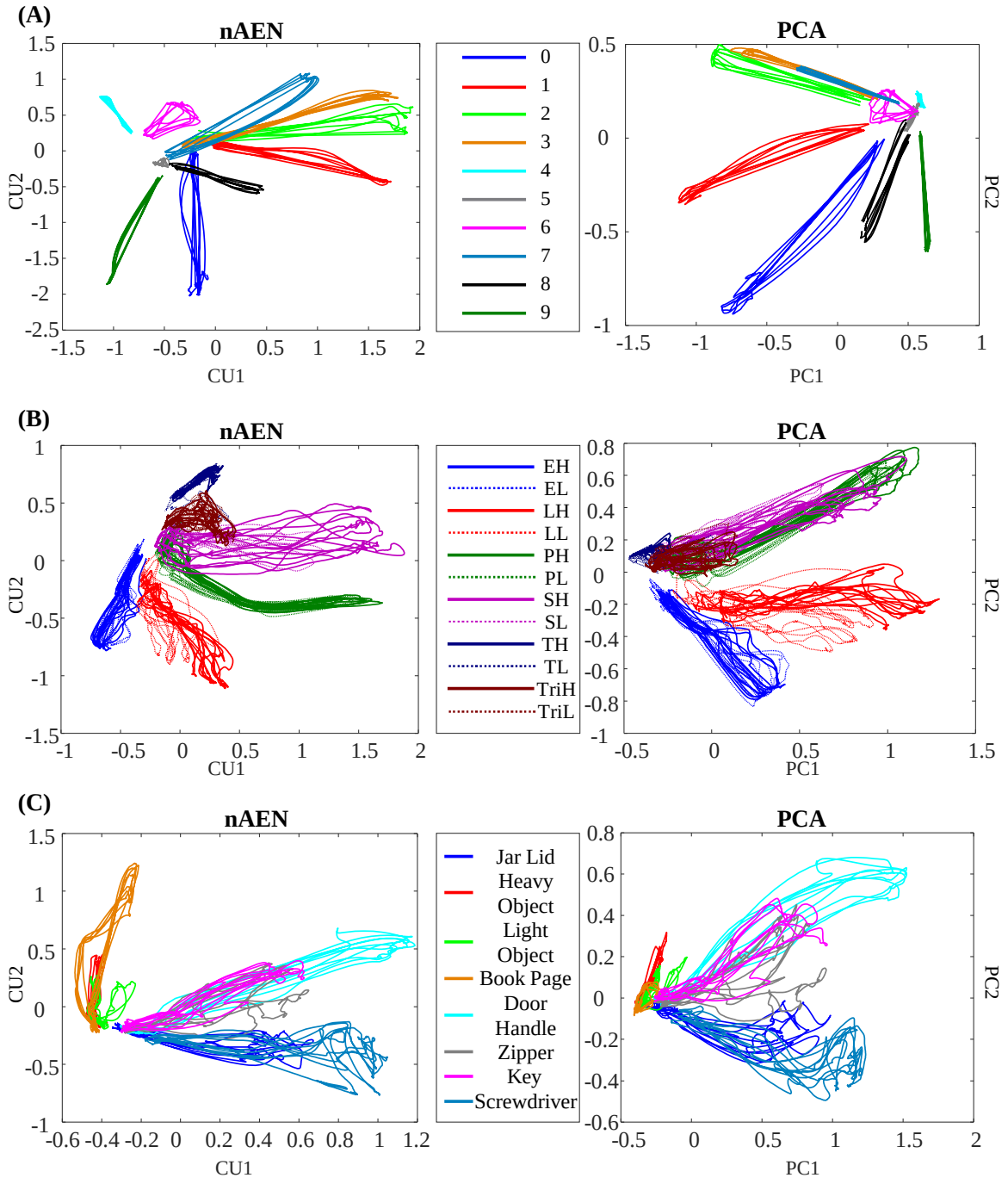

Figure 1.10 Visualization of the latent trajectories for nAEN and PCA for P1 performing (A) American Sign Language (ASL) Gestures, (B) Object Grasps, and (C) Activities of Daily Living (ADL) Tasks. The hand movements tested are represented with different colors. The latent manifold only consists of two dimensions. For Object Grasps, the legend can be interpreted as follows: the first letter represents the type of grasp (E, Extension; L, Lateral; P, Power; S, Spherical; T, Tip; Tri, Tripod); the last letter represents the weight of the object being grasped (H, Heavy; L, Light).

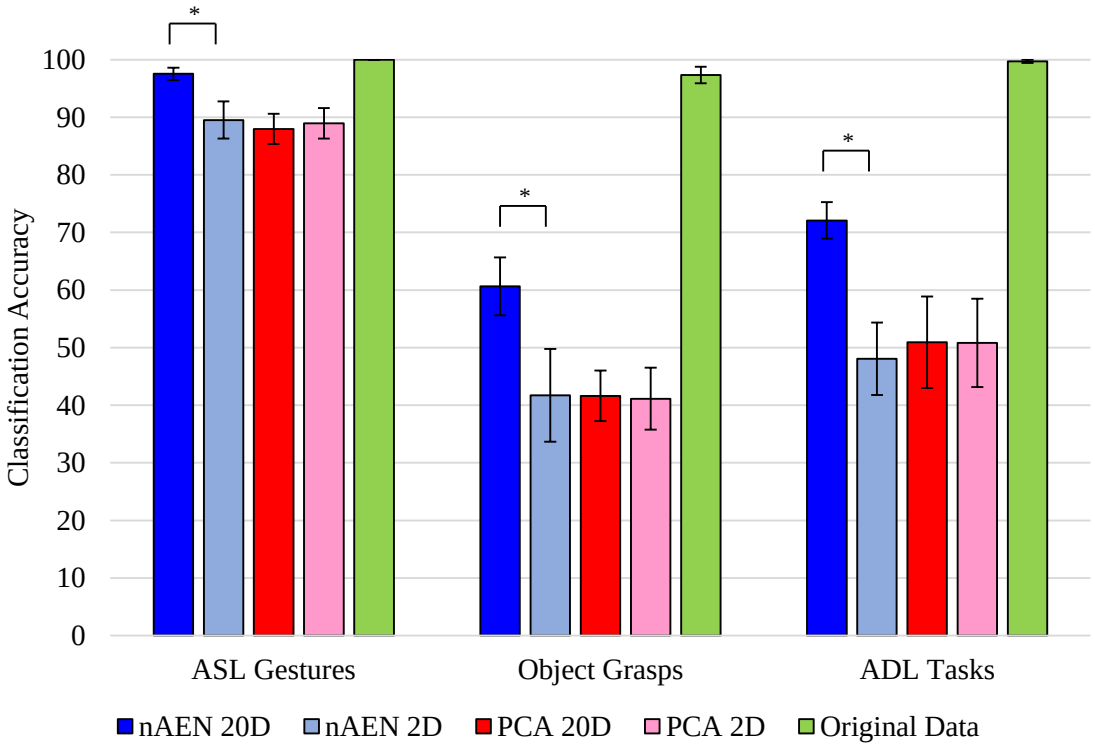

Figure 1.11 Accuracy of SoftMax regression applied to different datasets [American Sign Language (ASL) Gestures, Object Grasps, Activities of Daily Living (ADL) Tasks] across all participants. Regression was applied to original input data (green), reduced non-linear Autoencoder Network (nAEN) 2D (light blue) and reconstructed 20D (dark blue) data, as well as reduced Principal Component Analysis (PCA) 2D (light red) and reconstructed 20D (dark red) data. Error bars represent 95% confidence interval. Statistical significance: *indicates statistical significance of $p < 0.001$. No significant difference was found on the reduced 2D manifold of nAEN and PCA across all datasets. Original data exhibited high (nearly 100%) of movements. There was a significant difference between the reconstructed (20D) and reduced (2D) manifolds of nAEN with the former being more separable than the latter.

study, a non-linear equivalent of PCA, nAENs, demonstrated higher performance in: (i) reducing complex hand kinematics into a lower dimensional manifold with a smaller loss of data variability, (ii) creating higher spread of dimension variance in the latent manifold, and (iii) reconstructing a more separable manifold. All of these points could make nAEN a potentially effective at supporting continuous posture control for prosthetic hands.

1.4.1 Dimensionality Reduction

The ability to reduce the dimensions of kinematic data without ignoring their effective complexity is an essential yet challenging task to understand the biological mechanisms of control as well as to design precise artificial controllers. When it comes to developing a controller for multi-DOF hand prostheses, reducing the number of control signals may result in a more intuitive interface for the user. While the human brain is able to simultaneously manage multiple DOFs, such as those in the hands, often with high dexterity and precision, many research studies suggest that it does so through a smaller number of control "knobs" identified as synergies (Feldman and Latash, 2005; Santello et al., 1998; Ting and Macpherson, 2005; Todorov and Ghahramani, 2004; Weiss and Flanders, 2004).

To understand how dimensionality reduction may preserve the essential complexity of behavior, investigators have applied dimensionality-reduction techniques such as PCA to human hand coordination (Santello et al., 1998; Todorov and Ghahramani, 2004). Santello's group determined that 2 PCs were sufficient to account, on average, for $\sim$84% of hand kinematic variance, a higher value than the figure in our study. Such difference may be explained by the difference in the kinematic data. While the entire range of kinematic data from the REST to ACTIVE positions was recorded and used for analysis in our study, Santello's group only utilized one static instance of kinematics per imaginary grasp. In addition, the number of signals that was used in the Santello et al. was 15 whereas our study utilized 20 kinematic signals for analysis, which could have made it more difficult for PCA to reduce the dimensionality of the kinematic space.

Such dependence of PCA performance on the number of analyzed signals was also presented in the study of (Todorov and Ghahramani, 2004) as they ran PCA on 20 and 15 kinematic signals from a sensorized glove. They determined that in a 20- signal analysis, more PCs were required to account for the same variance of data than in a 15-signal analysis. In addition, they calculated that anywhere from three to seven PCs were required to account for 85% of data variance if the analyzed angle data was raw without normalization. The number of PCs depended on the performed task. In the aforementioned study by Todorov's

group, the analysis was performed on a larger set of kinematic samples, similar to our study; however, it was done on individual tasks rather than full datasets containing various tasks.

In our study, it was also noted that PCA performance was significantly inferior for ASL Gestures than for Object Grasps and ADL Tasks with just 2 PCs. Similar behavior could be observed in (Todorov and Ghahramani, 2004) study where the number of PCs to describe individuated joint movements was higher than that for object grasping and manipulation. This might be explained by the kinematic complexity of ASL gestures, in which independent joint control is required. In ADL tasks and object grasping, on the contrary, many joints move simultaneously to achieve the desirable posture or grasp, potentially joining into linear combinations that can be easily detected by PCA.

It is interesting to note that while PCA appeared to be less efficient with the ASL Gestures dataset (reconstructing with lower VAF), nAEN exhibited stronger performance with the ASL dataset (reconstructing with higher VAF) than with Object Grasps or ADL Tasks. This point was also clearly indicated in the inverse relationship between the signal variance of a dataset and the VAF with 2 CUs for nAEN. As a result, nAEN improved VAF and signal reconstruction when the input signal had more variability (as in ASL Gestures), contrary to the linear PCA. This is consistent with the observation that attempting to estimate the dimension of the data generated by a non- linear process with a linear method, like PCA, results in the overestimation of the actual process dimensionality (Tenenbaum et al., 2000). Accordingly, adding variance to the input data would result in the reduction of VAF by an insufficient number of PCs.

When comparing the performance of linear and non-linear dimensionality-reduction techniques, our study demonstrated that nAEN outperformed PCA by reconstructing over 90% of data variability with only 2 CUs. Such results overpower the dimensionality-reduction performance of PCA presented in earlier hand kinematic studies. A comparison of non-linear dimensionality reduction was performed earlier by (Romero et al., 2010) and by (Cui and Visell, 2014) over datasets obtained from hand grasping patterns. Somewhat surprisingly, Cui and Visell concluded that the quality of dimensionality reduction obtained by PCA was superior to that obtained by non-linear algorithms, including AEN. Our findings are not consistent with their conclusion. We believe that this discrepancy may be attributed to two factors. First, the analysis of Cui and Visell was limited to grasping, whereas our data set included other hand task. Perhaps, most notably, our data included ASL gestures and a broader spectrum of hand configurations associated with ADLs. In fact, in our dataset, the difference in performance between PCA and nAEN was smaller for hand grasps. A second observation concerns the performance measures. While we base our conclusions on VAF,

Cui and Visell adopted a criterion based on the preservation of neighborhood relations after dimensionality reduction. This criterion was based on Euclidean distance, which, as noted by the authors, has an implicit bias in favor of a linear method like PCA. And one can add that Euclidean distance is not a clearly applicable measure for angular manifolds. Like Cui and Visell, Romero and colleagues limited their analysis to grasping patterns. They compared the latent manifold generated by different non- linear dimensionality-reduction algorithms observing a better performance compared with PCA.

Our findings highlight the potential superiority of nAEN when used as a control method for hand prostheses. Higher VAF value signifies that with a smaller number of control signals, the nAEN-based controller would be competent to generate a more precise representation of multiple DOFs in a prosthetic hand, compared to PCA. As a result, tasks that require high precision and dexterity (*e.g.*, handcrafts, personal hygiene) may become more feasible. The decreased number of control signals required to control a precise motion of the prosthesis may result in lowering the mental burden on the users without sacrificing performance. For example, current techniques, such as PR in Coapt Gen2 (*Coapt LLC, Chicago, IL, USA*), rely on eight EMG signals to control a prosthetic device. The results of our study suggest that potentially by reducing the number of EMG signals to two and allowing each signal to control one CU would still allow the user to cover a large space of hand gestures.

The findings of this study also suggest the inherent non-linear nature of hand kinematics. The apparent differences between nAEN and PCA with only two dimensions in the latent manifold highlight the former's ability to capture components of the data that are not being picked up by the linear function of PCA. Evidently, this may suggest the need to, first, test and, potentially, utilize non-linear methods when analyzing biological systems, such as hand kinematics. While their mathematical manipulation might be more difficult and less intuitive, the importance of capturing non-linearities of a system might be of greater importance. We feel it is important to emphasize that our use of an artificial neural network, the autoencoder, is not associated with any claim or pretense to represent information processing in the neural system. This is presumably a limit shared by PCA, as there is no evidence that constraint of orthonormality is satisfied by neuromuscular activity. Nevertheless, a relevant element in our analysis is that the information that can be extracted from the observation of hand motions is better captured by a low-dimensional non-linear manifold than by a linear space. While there are other non-linear statistical methods for signal processing (Tenenbaum et al., 2000), AENs have the distinctive property of acting as non-linear filters whose parameters are set by training on an initial dataset. Then the network with these same parameters is used to project

incoming data on the same latent manifold of the initial dataset, under the critical assumption that the signal statistics has not changed.

1.4.2 Dimension Variance

Dimension variance is an important aspect of every controller. In some cases, the distribution of control authority across the controlled dimensions should be uniform as controlling each DOF may be similarly important. One of such cases includes control of a two-dimensional cursor on a screen-in Euclidean geometry, where all directions are by definition equivalent, the control of each dimension should be distributed equally. However, if the dimensionality reduction results in an uneven distribution (a large range of dimension variance), then control is distributed unevenly.

While keeping the number of control signals as low as possible is important in ensuring the mental load to the user is manageable, adding more signals can be crucial to allow the controller to account for important information that may otherwise be thrown away. As a result, it might not be useful to limit the controller to the minimal number of control signals.

In such cases, the dimensionality-reduction method of choice may be detrimental. From the results presented above, nAEN could prove its feasibility as a method that would allow developers adding control signals that account for task-relevant variability. In the case of PCA, if the controller requires six signals, the last two signals oftentimes have the variance of $< 5\%$ of the entire range of motion for all six dimensions. In most cases, such addition would be equivalent to adding noise. Noisy signals may decrease the controller performance and negatively affect the user's ability to perform tasks. In nAEN, on the contrary, adding more control signals does not organize data variance in a decreasing manner across dimensions but rather does so in a proportional manner where no signal accounts for $< 10\%$ of data variability.

With a much smaller difference between the dimensions of highest and lowest variance for larger number of dimensions, nAEN highlights its ability to distribute data information across CUs more evenly than PCA.

Another interesting point that was made in the analysis is the correlation between input data variance and the variance spread across latent dimensions in PCA. It appears that data of higher variance (*e.g.*, ASL Gestures) results in latent dimensions with a smaller range of variance, implying that variance is distributed more equally across dimension. On the contrary, when input data exhibits less variance (*e.g.*, Object Grasps, ADL Tasks), there is a large difference between dimensions of highest and lowest variance. Similarly, when VAF is

high, PCA appears to exhibit a more dramatic drop between dimension variance, thus proving this dimensionality-reduction method to be the less desirable choice even in cases when it exhibits high VAF. Such results highlight that in cases when PCA is able to reconstruct with a desirable VAF, the variance across its latent dimensions will not be equally spread, thus resulting in a higher chance of producing control signals that would appear noisier.

1.4.3 Latent Trajectories Visualization

Visualization of the latent trajectories can be useful in understanding the internal works of a dimensionality-reduction method. In addition, if this latent structure is utilized in the controller, it can aid researchers in identifying the properties of the control manifold.While visualization of the latent trajectories of PCA is simple due to the orthonormality of the principal eigenvectors and the overall linearity of the space they span, things are different with nAEN since CUs are neither orthogonal nor linear elements.

As one visualizes the latent trajectories of a nAEN by representing CUs as Cartesian coordinates, one must understand that this representation is unable to capture the non-linear properties of the manifold. However, research in visualization of non-linear manifolds has been limited.

In this study, the 2D Cartesian representation of the latent trajectories was used to visualize the separability of different movements within each dataset. From the results, it appeared that there was no significant difference between the separability of classes of nAEN and PCA manifolds across all subjects.

When visualized, certain movements appeared to be much closer in the 2D manifold to some than to others, increasing their chance of being misclassified. Such spatial closeness could be explained by the kinematic similarity of certain movements. For example, gestures 2 and 3 were closely placed on the 2D manifold of both nAEN and PCA. When examining the two gestures kinematically, one could notice that the only difference between the two was in the flexion of the thumb. Likewise, movements that were very different from each other kinematically (*e.g.*, gestures 1 and 9) appeared further away from each other on the 2D plot. Same results could be seen across other datasets, indicating that kinematic similarity resulted in closer appearance of the movements on the 2D manifold.

1.4.4 Movement Separability

Separability between nAEN and PCA in 2D was participant- and movement-dependent, implying that some participants created a more separable nAEN manifold for certain tasks

while others exhibited a more separable PCA manifold for the same tasks. Such occurrence may be due to the kinematic difference across participants performing various tasks. It is also important to note that the goal of the created nAEN structure was to minimize the error between the output and the input and not to create a more separable manifold of reduced dimension. However, one could rewrite the algorithm in such a way that the network would search for parameters that create a more separable manifold in the bottleneck layer.

It was noted in the results that separability of movements was much higher for ASL Gestures than other datasets for both nAEN and PCA. This could potentially be explained by hand kinematics being more distinct during ASL Gestures that clearly differentiate different classes. This is consistent with the very purpose of a sign language to generate readily distinguishable patterns. In addition, it is important to note that both heavy and light variations of the same grasp type were used as separate classes in Object Grasps, which could make it more difficult for the classifier to differentiate between them since they were very similar kinematically. If other information, such as for example, kinetic, was used in addition to kinematic data to differentiate between different grasps, the classification accuracy of Object Grasps would potentially be higher. Lastly, in ADL Tasks, many movements required similar grasping types, which, in turn, resulted in similar kinematic output, making classes less differentiable.

While nAEN exhibited an increase in separability of classes when going from the 2D latent manifold to its 20D embedding, no such difference was observed for PCA. The latter result is expected, because with PCA the latent manifold is a 2D plane embedded in the 20D dimensional signal space. In this linear case, the Euclidean distances between points in the plane are the same if we take them over the plane or over the embedding signal space. The same cannot be concluded with non-linear dimensionality reduction, because the latent manifold is now a curved space and distances between points over a curved surface are generally different when taken over the surface or over the embedding space. In fact, distances over a curved surface (think of a sphere) can only be longer than the differences over the embedding signal space. The results in Figure 1.10 show that this difference in Euclidean metric leads to a poorer classification when the distances are taken in the local coordinates of the latent manifold, as points belonging to the same class are mislabeled as belonging to different classes. Figure 1.10 also shows that there is not a difference in classification accuracy between nAEN and PCA, when the data are taken in the respective low-dimensional latent representations. Therefore, we do not have a case for using the non-linear rather than linear dimensionality reduction for a prosthetic controller based on PR. However, the conclusions are different for a prosthetic system based on continuous control,

where the reconstruction error and the variance accounted for play a greater role and where these both best captured by the non-linear dimensionality reduction (Figures 1.5, 1.6A).

An interesting note could be made regarding the high separability of classes in the original input data. It is important to understand that the original data contains 20 signals that can each vary across different movements. And although every movement started from approximately the same position in every dataset, differences in signals that could appear insignificant on their own could result in a significant difference when added together across all 20 signals. Hence, it is intuitive that classes are more separable when more dimensions are present. However, in such a case, both PCA- and nAEN-reconstructed 20D spaces should exhibit just as high accuracy. While we noted before that the PCA-reconstructed space might not see a significant improvement in class separability when going from 2D to 20D, a careful consideration of the nAEN results must be made. Although there was a significant improvement when going from 2D to 20D for nAEN, the 20D results were still not as high in accuracy as those of the original input data. This could be explained by the potential elimination of insignificant signal differences during reconstruction by the nAEN. These insignificant differences could be taken by the autoencoder as noise, which it aimed to reduce in the system, only leaving information that produced high variability. As a result, data with small variability is not reconstructed. This observation can also lead to the conclusion that PCA, when reconstructing, removes more of low-variability samples from the data, resulting in a less separable reconstructed space.

1.4.5 Other Applications in Prosthetic Control

It is important to mention that the findings of this study might have wider applications beyond the myoelectric control. One of such examples would be in hardware development similar to the aforementioned “Soft Hand”, in which the simplified prosthetic control relies on the linear combination of the first n PCs (Della Santina et al., 2017). This results from the linear nature of PCA that is discussed earlier in this paper. To obtain the desired posture in a hardware using nAEN, one might utilize the decoder part of the network and pass the first n CUs through the third and fourth layer as described in Equations 1.3 and 1.4. In such way, the development of the hardware of the prosthetic hand would rely on the composition of the decoder component of nAEN.

In addition, when dealing with myoelectric control for prosthetic hands, one must consider not only the domain of kinematic behaviors of the device, but the control signals themselves (*i.e.*, EMG). While reducing the dimensionality of kinematics is of high importance, one

can consider applying dimensionality-reduction methods studied in this paper to the control signals. This would expand the potential of using a greater number of signals, fused in a latent manifold, for prosthetic control. Such application of nAEN will be explored with EMG signals in future research of improving myoelectric control of hand prostheses.

1.4.6 Supervised vs. Unsupervised Learning

Many of the state-of-the-art techniques in prosthetic control involve supervised learning methods. For example, the PR method utilizes a form of a classification algorithm, which typically consists of a feature extraction from the given EMG signal and feature classification of the desired hand movement (Geethanjali, 2016). With the use of this technique, prosthetic users are able to associate certain EMG patterns with desired grips, thus decreasing the time it takes to select and perform the anticipated movement. Despite of its rising popularity, PR control exhibits issues such as long training time, chances of inaccurate classification, and being limited to a finite number of preselected hand postures (Atzori et al., 2016; Castellini et al., 2014; Geethanjali, 2016; Hargrove et al., 2006; Scheme and Englehart, 2011; Young et al., 2011).

In addition, all forms of supervised learning are limited by their dependence on labeled data, the ground truth, based on which they learn a function that best approximates the relationship between the input and output observable in the data. For unimpaired individuals, creating supervised output might not be an issue (*e.g.*, flex a joint to a specific degree, etc.). However, for individuals with motor impairments, such task is inherently difficult or completely unachievable, making the creation of labeled data impossible.

In contrast, PCA and AENs are unsupervised algorithms, whose data do not need to be labeled as their goal is to learn the data's statistical properties rather than minimizing some classification error. As a consequence, a user interface based on unsupervised methods can adapt to the particular statistics (*e.g.*, in kinematics) of the users without requiring them to perform specific movements in specific ways.

Another form of supervised learning is regression, which some research groups have proposed for prosthetic control (Geethanjali, 2016; Muceli and Farina, 2011; Muceli et al., 2013; Ngeo et al., 2014). This is a form of a learning method, in which the output data is continuous in contrast to that of a classifier. While it may be a useful feature in the development of a continuous controller, in which device movements are not limited to a number of preset postures, regression has not been proven effective in its use with prostheses. Oftentimes, regression algorithms developed for prosthetic hands require clean forearm EMG

signals that can be associated with certain hand movement. However, in case of transradial (below-elbow) amputations, these signals are heavily dependent on the site of amputation, amount of residual limb, and many other factors (Li et al., 2010). Clean surface EMG signals can be obtained in a lab setting from able-bodied individuals, but are much harder to get from amputees outside of the lab, thus making regression less effective control tool for prostheses.

In summary, unsupervised continuous learning methods, such as nAENs, promise to be a useful tool in the development of prosthetic controllers in addition to their superior performance in dimensionality reduction.

1.4.7 Supplementary Materials

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.338 2020.00429/full# supplementary-material

1.5 Appendix

Other than the kinematic signals discussed in the main body of the manuscript, we also compared the performance of PCA against that of nonlinear AE in estimating the dimensionality of muscular signals.

We used a Trigno Wireless System (*Delsys INC, Natick, MA, USA*) to acquire the electromyographic (EMG) signal from eight upper body muscles. We placed the sensors along four muscles: biceps brachii, lateral triceps, medial deltoid and upper trapezius, both from left and right part of the body. We recorded the raw EMG signal coming from each sensor with a sampling rate of 1111.11 Hz. The raw signal was preprocessed with a series of 2nd order IIR Biquadratic Butterworth filters to obtain EMG's envelope. The raw EMG signal had been bandpass filtered ($f_c = 30 - 450Hz$) to remove high frequency noise and low frequency artifacts caused by body motions, rectified and finally low-pass filtered ($f_c = 2Hz$). The EMG envelope was finally downsampled at 50 Hz.

We enrolled 9 subjects (age 26.0 ± 2.2) with no known history of motor impairment. They performed upper body-movements for 2 minutes during which they explored freely their range of motion without performing extreme and uncomfortable movements. In this phase, EMG data were collected and used as input signals for PCA and nAEN.

Similarly to the analysis on the kinematic signals in the main body of the manuscript, we compared the performance of PCA and nAEN algorithms with two, three, four, five, and six latent dimensions.

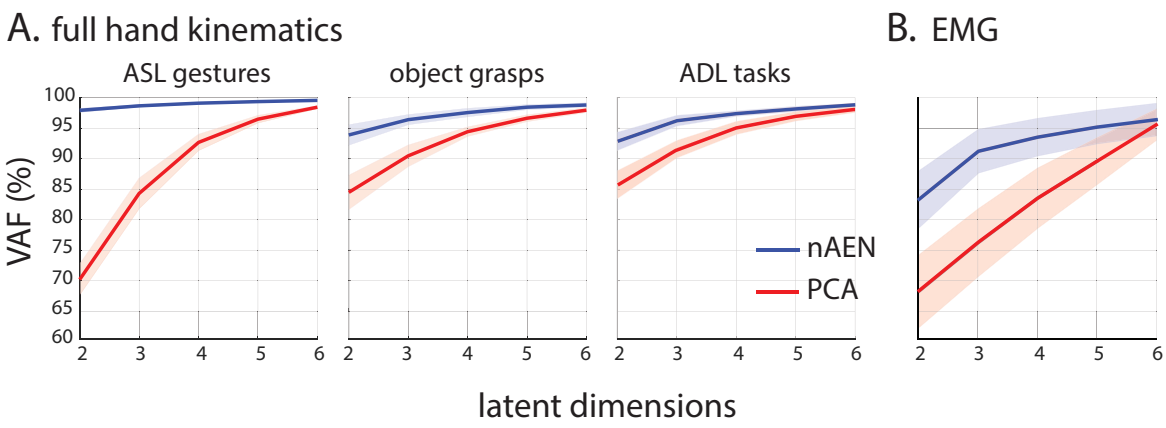

Figure 1.12 Performance of non-linear Autoencoder Network (nAEN, blue) vs. Principal-Component Analysis (PCA, red) with 2, 3, 4, 5, and 6 dimensions in the bottleneck layer on three kinematic datasets (Panel A: ASL gestures, object grasps, ADL tasks) and the EMG dataset Combined (Panel B). The performance is calculated using Variance Accounted For (VAF). Solid lines represent average results across all paritcipants. Faint shaded lines represent 95% confidence interval. nAEN outperformed PCA across all datasets and for all number of latent dimensions tested in this study. For each latent dimension, the VAF of EMG signals was significantly lower than that of any of the kinematic signals.

For the EMG dataset, nAEN outperformed PCA by reconstructing the input data with higher VAF for two, three, four, five, and six latent dimensions (Figure 1.12B). The difference was more remarked for two (nAEN:$\sim$ 85%, PCA:$\sim$ 70%), three (nAEN:$\sim$ 90%, PCA:$\sim$ 75%) and four (nAEN:$\sim$ 95%, PCA:$\sim$ 85%) latent dimensions, and decreased as the number of dimensions in the latent manifold increased.

Interestingly, the VAF of the EMG dataset was consistently lower than that of any of the kinematic signals. To reach 90% of VAF in the EMG space, three latent dimensions were needed with nAEN (while two were enough to account for 95% of the original variance of each kinematic signal). With PCA, this difference was even more evident, since the 90% VAF threshold was only reached with a 5-dimensional latent space.

These results are an additional evidence of the superiority of nonlinear DR methods in estimating the dimensionality of body signals. The kinematic and EMG signals considered here were not correlated, since they were collected at different times and on different body segments (wrist vs upper body). However, the results suggest that encoding muscular signals might require more latent dimensions than in the case of kinematic signals.

Chapter 2

Cartograms: a novel approach to the analysis of high-dimensional signals

2.1 Introduction

Through the centuries, cartographers have developed several ways to faithfully represent the three-dimensional structure of paths over the earth over two-dimensional maps. In our times, flying over the Atlantic Ocean we can comfortably monitor on a small screen the route followed by our aircraft. This path is invariably curvilinear, apparently violating the concept that the shortest distance between two points is a straight line. All of this of course is just the consequence of our planet being approximately a sphere immersed in a three-dimensional space. While this external space is approximately Euclidean, the surface of the earth is not; its curvature is demonstrated by the violation of a few cardinal rules, such as the angles inside a triangle adding to 180 degrees and two parallel lines never meeting. The cartographers provide us with a faithful representation of the geometry of the earth by drawing its two-dimensional charts in ways that reflect the "true" three dimensional embedding space. Similar efforts are also of value for neuroscientists who want to make sense of tens or hundreds of signals collected from multiunit neural recordings or from sensors detecting the movements of multiple body segments. Our visual system can at most represent two dimensional projections of three-dimensional images. While we cannot see

The content of this chapter has been prepared for submission as Rizzoglio, F., De Santis, D., Portnova-Fahreeva, A., Solla, S., Casadio, M. and Mussa-Ivaldi, F.A. 2020. Cartograms: a novel approach to the analysis of high-dimensional signals.

a neural trajectory in ten dimensions, we can lower the dimensionality of this trajectory by established methods that reduce the description to a lower dimensional "latent" representation by removing linear or nonlinear relationships between the original dimensions. We can thus obtain two- or three-dimensional representations of ten-dimensional signals containing as much information as possible for reconstructing the original data. Once we have derived such low-dimensional representations, we can visually represent the data as curves or surfaces by applying the same methods developed by the cartographers. Our purpose here is to explore the limits and potentials of such cartographic approach.

Low-dimensional *latent* manifolds that preserve the information of high-dimensional neural signals have been derived and analyzed (Cunningham and Byron, 2014; Gao and Ganguli, 2015; Pandarinath et al., 2018a; Sadtler et al., 2014; Stopfer et al., 2003). Latent manifolds with dimension $\leq$3 allow for visual inspection of their shape and for assessing properties like local curvature, thus improving the interpretability of the original data structure (Keim, 2002). As an example, Gallego and colleagues unveiled the existence of neural manifolds and represented on a Cartesian plane the patterns of neural variability expressed on these manifold (Gallego et al., 2017, 2018). Representation of the activation of neural modes, the latent trajectories (Yu et al., 2008), has been accomplished by Principal Component Analysis (PCA) (Wold et al., 1987), a dimensionality reduction (DR) technique that projects data onto a linear subspace with minimum loss of variance. Although linear DR algorithms such as PCA are effective at finding a two- or three-dimensional latent manifold that can be readily visualized, they tend to overestimate the dimensionality of data that lie on non-linear manifolds (DeMers and Cottrell, 1992; Tenenbaum et al., 2000). Non-linear DR techniques based on autoencoder (AE) networks (Kramer, 1991)) have seen a recent surge of interest (Farshchian et al., 2018; Pandarinath et al., 2018b), as they allow to obtain non-linear manifolds that retain more covariance of the original data than their linear counterparts (Portnova-Fahreeva et al., 2020). However, the interpretability of such methods is not straightforward, as their learning dynamics and mechanisms that underlie the found solutions are difficult to grasp (Fan et al., 2020; Vellido et al., 2012).

In this study, we propose a visual tool to unveil the hidden processes that take place during AEs training via the transformed display of its latent manifold. This display provides insights about the structure of the nonlinear latent manifold and about the information retained during the AE training, thus allowing for a direct assessment of the extent to which the nonlinear analysis provided by the AE does or does not outperform the analysis provided by linear methods. The type of latent visualization developed here gives us a heuristic

intuition of geometrical features of the input space that are usually cumbersome to identify for multivariate signals (Vellido et al., 2013).

Visualization of non-linear manifolds is a non-trivial endeavor, as their non-Euclidean nature requires accounting for distortions and warping. Latent trajectories derived from AEs are typically represented in a Euclidean coordinate system without corrections for their intrinsic nonlinearities (Portnova-Fahreeva et al., 2020). To overcome this limitation, we adopt an algorithm of Multidimensional Scaling (MDS) (Gianniotis, 2013) that provides direct information about the non-Euclidean aspects of the geometry of latent signals. The distortions of the manifold are captured by requiring that the Euclidean distances between points in the embedding space correspond as closely as possible to the distances between their decoded projections within the manifold. This requirement results in a *cartographic correction* to the nonlinearities detected by the AE.

We tested the MDS based approach on a variety of synthetic datasets commonly used as a benchmark for manifold learning techniques (Gashler et al., 2008; Silva and Tenenbaum, 2003). As the true geometrical structure of these datasets is known, we can identify which aspects of this geometry are preserved and which are lost by inspecting the transformed low-dimensional manifold. The work is organized as follows.

We started by evaluating the effect of training an AE with different parameters on the reconstruction performance of the resulting AE and on the shape of the latent manifolds. The training of an AE of a specific architecture with a given training set can result in different solutions (Liao and Poggio, 2017) due to differences in initial conditions, learning rate, stopping criteria, and other hyperparameters that are typically chosen so as to mostly minimize the reconstruction error (Claesen and De Moor, 2015). Here we investigated how the choice of hyperparameters affected the shape of the latent manifold, particularly its local curvature, and whether these potential effects were correlated with changes in reconstruction performance. In addition, we compared latent representations before and after the cartographic correction to test whether the cartographic correction might identify nonlinear aspects of the dataset geometry captured by the manifold.

We then addressed the issue of characterizing the differences between manifolds that result from training a specific AE architecture with a given training set when using different hyperparameters. An alignment procedure allowed us to establish the uniqueness of this manifold, a discovery that relies on the cartographic correction.

Finally we monitored the evolution of both latent manifolds and reconstructed signals during the training of the AE. The changes in manifold nonlinearities provided by the

cartographic correction offers a graphical account of the process through which nonlinear AEs outperform linear DR techniques.

2.2 Methods

2.2.1 Autoencoder networks

AEs are artificial neural networks capable of learning efficient lower representations $\in \mathbb{R}^m$ of higher dimension data $\in \mathbb{R}^d$ without supervision, with $m < d$. Importantly, unlike other nonlinear methods for DR (*e.g.*, Isomap (Tenenbaum et al., 2000) or t-SNE (Maaten and Hinton, 2008)), after being trained on a dataset, AEs can be used to project new incoming data on the manifold obtained from the training data. In an AE, the information flows from the d-dimensional input layer through layers of connected neurons (hidden layers) to the d-dimensional output layer. Each neuron processes its inputs through a linear combination characterized by weights w and biases b, followed by a scalar activation function that applies an additional linear or nonlinear transformation. An AE that uses linear activation functions implements PCA (Baldi and Hornik, 1989)(Plaut, 2018).

An AE is composed of two parts: an encoder E that converts each d-dimensional input q input to an m-dimensional latent representation, the *code*, followed by a decoder D that converts the latent representation into the output, a reconstruction $\hat{q}$ of the input. An AE works by approximating the identity map that copies q to $\hat{q}(\{w\},\{b\})$ with minimum loss, thus learning an efficient internal representation of the input within the latent manifold. Here, $\{w\}$ and $\{b\}$ respectively represent all the weights and biases connecting neurons within the network. The AE is trained through the minimization of the mean squared error between q and $\hat{q}$:

$$J(q,\hat{q}) = \frac{1}{P}\sum_{i=1}^{P}(\hat{q}_i(\{w\},\{b\}) - q_i)^2 \tag{2.1}$$

where P is the number of points in the training dataset. We focused on AE networks with an $m = 2$ latent manifold, and trained AEs with both linear and non-linear activation functions, as described below.

2.2.2 The AE latent manifold transformed via MDS

A scheme of the proposed approach is depicted in Fig. 2.1. The problem of visualizing a latent manifold $L \in \mathbb{R}^m$, that represents data $q \in \mathbb{R}^d$, with $m < d$, consists in finding a mapping f:

$$f : \mathbb{R}^d \to L \tag{2.2}$$

Here, we consider a 2D latent manifold ($m = 2$) for its immediacy of visualization and to lower the computational cost. Although conceptually the approach can be extended to any m, computational costs make it unfeasible to apply it beyond $m = 3$. Moreover, the ability to visualize the manifold would be lost for $m > 3$. An AE trained on data $\{q\}$ provides an implementation of the map f through the encoder subnetwork E. However, if the resulting encoded points $l \in L$

$$l = E(q)\,, \tag{2.3}$$

were to be visualized in a Cartesian plane, a mathematical inaccuracy would be introduced, as the nonlinear nature of E must result in deviations from Euclidean geometry in the set $\{l\}$. We will refer to $\{l\}$ as the encoded points in the *uncorrected latent manifold*. Here, we adapted a procedure described in (Gianniotis, 2013) to evaluate and correct for the non-linearities embedded in the AE, obtaining a *corrected latent manifold* L_c

$$l_c = g(l)\,, \tag{2.4}$$

where $\{l_c\}$ are now the encoded points in the corrected latent manifold.

To implement the mapping g, we first constructed a rectilinear coordinate chart by superimposing a C by C grid over the 2D uncorrected latent manifold. The size of the grid was prefixed and its position within the uncorrected latent manifold was chosen such that it was centered on the mean of l. The nodes of the grid are denoted by $x_j \in \mathbb{R}^m$, $j = 1, \cdots, C^2$.

Then, the AE decoder D was applied to the grid to obtain its embedding M, with nodes $D(x_j) \in \mathbb{R}^d$. M is a $\mathbb{R}^m$ manifold embedded in the $\mathbb{R}^d$ space of the data. Due to the nonlinearities of the AE, the manifold M could stretch and contract in order to capture the data density. In spite of these nonlinearities, we expect the manifold to be locally flat and thus describable by Euclidean geometry. Thus if two points x_k and x_j were neighbors we computed a local distance A_{kj} by measuring the L_2 distances between these two points in $\mathbb{R}^d$:

$$A_{kj} = \begin{cases} \|D(x_k) - D(x_j)\|_2, & \text{if } x_k, x_j \text{ neighbors} \\ \infty, & \text{otherwise} \end{cases} \tag{2.5}$$

This procedure results in a C^2 by C^2 matrix A of local distances. The matrix R of global distances follows from computing the geodesic between each pair of decoded nodes points $D(x_k)$ and $D(x_j)$; it was obtained by applying the Floyd-Warshall algorithm (Aini and Salehipour, 2012) to A. Finally, the original grid defined in the uncorrected latent manifold was redrawn by displacing all the nodes so that all pairwise distances optimally match as well corresponding high dimensional distances specified in R. The corrected locations of the new set of nodes $\{\xi\}$ were found by minimizing the cost function:

$$S(\{ł_c\}) = \sum_{k=1}^{C^2} \sum_{j>k}^{C^2} \|\hat{R}_{kj} - R_{kj}\|_2 \tag{2.6}$$

where $\hat{R}$ is defined as the matrix of Euclidean distances between pairs of nodes in $\{\xi\}$. Hence, Euclidean distances between nodes on the distorted grid reflect geodesic distances between high dimensional data lying on the manifold M. The newly distorted grid, the *cartogram*, contains information about manifold nonlinearities such as curvature, and it provides a metrically accurate (Euclidean) reference system. This is the corrected latent manifold.

The procedure that implements 2.6 results in the specification of corrected locations for all nodes on the grid selected to cover the uncorrected latent manifold. If additional points were to be incorporated, the MDS algorithm would need to be fully reoptimized. Moreover, increasing the density of the grid so as to capture manifold nonlinearities with greater resolution would incur in a huge computational cost, which scales exponentially with the number C of grid nodes per latent dimension. To circumvent these limitations, we trained an artificial neural network (ANN) (Hornik et al., 1989; Liu et al., 2017) to approximate the function g defined in Eq. 2.4. The ANN effectively learnt a function $\widetilde{g} \sim g : L \to L_c$ to map points on the uncorrected latent manifold onto the corrected one. The ANN structure consisted of two hidden layers with 20 neurons each and used a scaled exponential linear unit (selu) as activation function (Klambauer et al., 2017). Inputs to the ANN were the locations of the grid nodes x_k on the uncorrected latent manifold, and the outputs were the nodes ξ_c in the corrected latent manifold. The ANN learned to minimize the mean squared error between x_k and ξ_k:

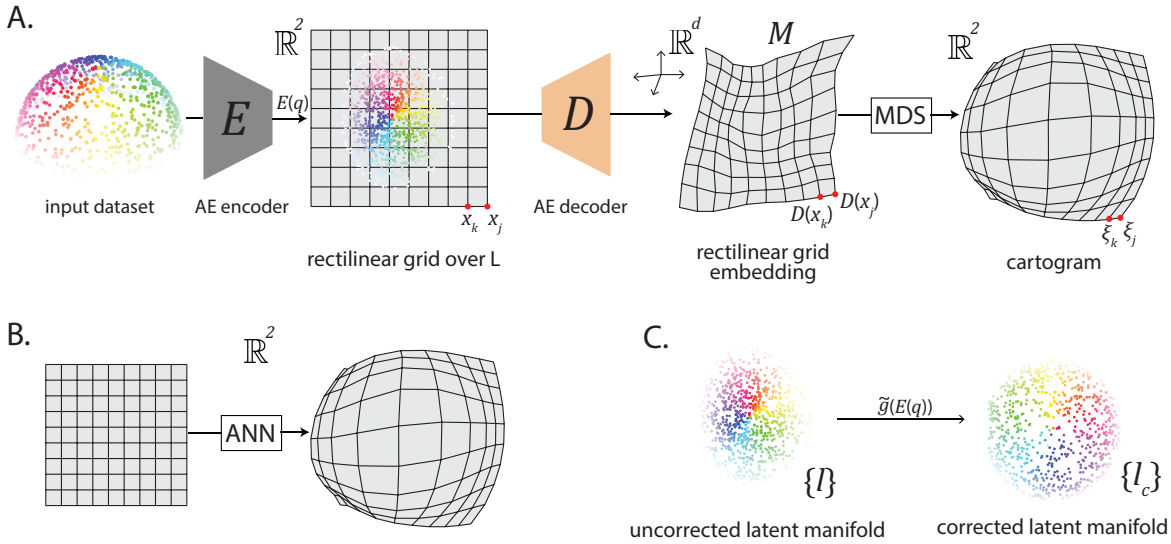

Figure 2.1 Method for obtaining the corrected 2D latent manifold using a combination of an AE, MDS, and an ANN. **A**. The dataset $\{q\}$ provides inputs to the E module of the AE that outputs points $\{x\}$ in the uncorrected latent manifold L. A rectilinear grid is superimposed on L and projected back in the high dimensional data space using the D module of the AE to obtain the embedded manifold M. MDS is then used to reshape the rectilinear grid as a *cartogram* that preserves distances along M. **B**. An ANN is trained to approximate the function g, using as inputs the locations of the grid nodes on the uncorrected latent manifold and as outputs the nodes of the cartogram. **C**. The resulting function $\widetilde{g}$ projects points on the uncorrected latent manifold onto the corrected one.

$$\widetilde{g} = argmin\left[\frac{1}{C^2}\sum_{k=1}^{C^2}(x_k - \xi_k)^2\right], \tag{2.7}$$

and the complete equation to map the data into the corrected latent manifold is:

$$l_c = \widetilde{g}(E(q)) \tag{2.8}$$

2.2.3 Synthetic data sets

For the purpose of this work we made use of various nonlinear datasets typically used for benchmarking manifold learning techniques (Gashler et al., 2008; Silva and Tenenbaum, 2003).

First, we tested our AE based method's capability of modeling a *swiss roll* dataset (Fig. 2.2, left) (Pedregosa et al., 2011). The swiss-roll dataset contained 2000 points, colored

Figure 2.2 Synthetic datasets. From left to right: swiss roll, fishbowl, hemisphere.

according to their location on the manifold. Since the points on the swiss roll lay on a two-dimensional curved manifold, we expected a nonlinear AE with two code units to account almost fully for the variance of the dataset.

To further test the AE based method's ability to model curved manifolds whose intrinsic geometry is non-Euclidean, we considered two datasets of points lying on the surface of a three-dimensional sphere. Specifically, we randomly sampled 2000 points from a uniform distribution in a circular disk and then projected stereographically (hence, conformally mapped) onto a sphere. The two datasets were then obtained by horizontally cutting the sphere at two different heights. First, we cut the sphere with a plane located somewhere between the equator and the south pole and refer to the resulting data set as *fishbowl* (Fig. 2.2, center). Finally, we cut the sphere with a plane located at the equator level and obtained the *hemisphere* (Fig. 2.2, right). Since both sphere-based datasets are intrinsically two-dimensional, we again expect a nonlinear AE with two code units to be able to account for their variance.

2.2.4 Description of the three studies

Study I: Effects of AE training (hyper)parameters on reconstruction performance and latent visualizations

We evaluated the effect of training AEs with different architectures, total number of parameters, type of activation function, and initialization, on the reconstruction performance and the latent visualization of each AE for each of the three synthetic datasets described in §2.2.3. Each dataset consisted in 2000 points, split into 80% of training points (1600) and 20% of test points (400).

To evaluate how the number of parameters in the AE impacted training performance, we considered a condition in which the AE was *under-parametrized* with respect to the number of training data points and another in which the AE was *equally-parametrized*. For the

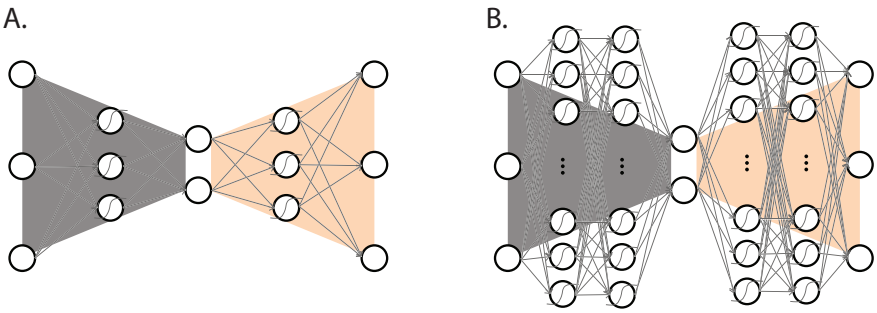

Figure 2.3 **A**. Under-parametrized autoencoder network trained on a synthetic dataset. The 3D inputs are fed to a 3D hidden layer and then to a 2D latent manifold. **B**. Equally-parametrized AE network. The 3D inputs are fed to two subsequent 25D hidden layers (not all neurons are shown for clarity) and then to a 2D latent manifold. The decoder subnetwork of both autoencoders is a mirrored version of the respective encoder subnetwork.

under-parametrized network, we included two hidden layers, one in the encoder subnetwork and one in the decoder subnetwork, with three neurons per layer, for a total of 41 parameters (Fig. 2.3A). For the equally-parametrized network, we increased the number of hidden layers to four, two in the encoder subnetwork and two in decoder subnetwork, and the number of neurons per hidden layer to 25, for a total of 1605 parameters (Fig. 2.3B). Both networks implemented a DR of the data via a two-dimensional latent manifold.

To assess the impact of utilizing a linear or nonlinear AE architecture, as well as different types of activation function, we trained equally-parametrized AEs with a linear (Baldi and Hornik 1989) and three commonly used nonlinear activation functions, namely: hyperbolic tangent (*tanh*) (Kalman and Kwasny, 1992), rectified linear units (*ReLU*) (Agarap, 2018), and scaled exponential linear units (*selu*) (Klambauer et al., 2017). The initial AE weights were chosen following the Xavier normal initialization (Glorot and Bengio, 2010). To ensure consistent initializations, we set the random seed that initialized the pseudo-random number generator for the AE weights to be the same for each AE training. The initial biases were set to zero. Finally, to evaluate the impact of different initializations, we compared the performance of an equally-parameterized AE architecture initialized with 20 different seeds. We repeated the procedure for each of the four types of activation function.

We assessed the ability of each AE to reconstruct the variance of their input signals via the *Variance Accounted For* (*VAF*), defined as:

$$VAF = \left(1 - \frac{var(\{q - \hat{q}\})}{var(\{q\})}\right) * 100\,, \tag{2.9}$$

where $\{q\}$ and $\{\hat{q}\}$ are the original and the reconstructed synthetic dataset, respectively. Higher values of VAF imply a larger amount of information about the dataset after the dimensionality reduction implemented by the latent manifold. In addition, we projected the synthetic datasets onto both the uncorrected and the corrected latent manifolds, and visually inspected their shapes for nonlinearities.

We expected the impact of each (hyper)parameter to be different on the VAF and on the nonlinearities of the latent manifold. For instance, while a higher number of AE parameters should trivially yield a higher VAF, the effect of different activation functions and/or initialization is more complicated to determine a-priori. The minimization of the reconstruction error that implements the training of the AEs does not impose any condition on the topological organization of the encoded data. Thus, the resulting uncorrected latent manifold is not expected to capture information about the metrics of the data. By contrast, we designed the cartographic correction to enable the AE to maintain the metrics of the synthetic dataset within the corrected latent manifolds.

Study II: discovering a unique latent manifold via cartographic correction

We investigated the possible existence of a *unique* latent manifold by aligning latent manifolds obtained for the same dataset but with different AE parameters. A unicity of the latent manifold would imply that the AE recovered the true manifold embedded in the dataset, independently of the choice of hyperparameters and the particular solution that the trained AE converges to.

In order to test for this possibility, we devised a method for aligning two latent manifolds L_1 and L_2 corresponding to the same dataset but encoded by two different AEs. We compared the case of aligning uncorrected manifolds with that of aligning corrected manifolds and evaluated the differences between these two cases. The hypothesis is that the cartographic correction would reduce the differences between the two latent manifolds, hence facilitating the discovery of a unique manifold. The first step is to center the data in both L_1 and L_2 at zero, and then to scale each latent manifold by its maximum absolute value. We then implemented a rotation and a mirroring operation in L_2 so as to obtain new latent coordinates $\widetilde{L}_2$ that align as well as possible with L_1. This is achieved through the minimization of a cost function that measures the residual distance between corresponding encoded points l_1 and $\widetilde{l}_2$ following a rotation and mirroring operation:

$$O(\theta, G_x, G_y) = \frac{1}{P}\sum_{i=1}^{P}\left(l_1^{(i)} - \widetilde{l}_2^{(i)}\right)^2 =$$

$$= \frac{1}{P}\sum_{i=1}^{P}\left(l_1^{(i)} - \begin{bmatrix} sign(G_x) & 0 \\ 0 & sign(G_y) \end{bmatrix}\begin{bmatrix} cos(\theta) & sin(\theta) \\ -sin(\theta) & cos(\theta) \end{bmatrix} l_2^{(i)}\right)^2, \tag{2.10}$$

where P is the number of points in the training dataset, θ defines the rotation of $\widetilde{L_2}$ relative to L_2, and G_x,G_y define the mirroring operation on the two coordinates of L_2. The optimization was constrained to rotations and mirroring to preserve the structure of data covariance in L_2.

To quantify the similarity between the aligned latent manifolds, we preserved the zero mean and the empirical covariance matrix of each latent encoding, and approximated each 2D probability distribution by a Gaussian: $p_1(l_1) \sim N(l_1; 0, \Sigma_1)$ and $p_2(\widetilde{l_2}) \sim N(\widetilde{l_2}; 0, \Sigma_2)$; this assumption chooses the maximal entropy distribution that preserves first and second order moments. The Kullback-Leibler Divergence (KLD) (Kullback and Leibler, 1951) between these two distributions,

$$D_{KL}(p_1, p_2) = \frac{1}{2}\left(tr(\Sigma_2^{-1}\Sigma_1) - 2 + ln\frac{|\Sigma_2|}{|\Sigma_1|}\right) \tag{2.11}$$

was used to quantify their dissimilarity. Since the KLD is not symmetric, we report both $D_{KL}(p_1, p_2)$ and $D_{KL}(p_2, p_1)$. Lower values of KLD indicated higher similarity between latent manifolds. As the distribution of data within each latent manifold can vary as the parametrization of the corresponding AE changes, we expected that directly comparing latent encodings would lead to inconclusive results. We hypothesized that two latent manifolds could be better compared after applying the cartographic correction that accounts for common input metrics. To verify this hypothesis, we computed the KLD for all available pairs of latent either uncorrected or corrected latent encodings after training an equally-parametrized AE using each of the three nonlinear activation functions described in §2.2.4 for each input dataset. To evaluate the effect of different AE initializations, we compared the latent manifold derived from the equally-parametrized AE with seed 0 to those obtained using the other 19 seeds, for a total of 19 KLD values for each dataset and nonlinear activation function, both before and after the cartographic correction.

Study III: studying the mechanisms of autoencoder learning

Lastly, we exploited the possibility of latent manifold visualization offered by the cartographic transformation to observe how the AE latent representation of the input data varied across training iterations. We trained a cartogram at regular intervals throughout the AE training (every 10 training epoch – until the 100th epoch – and then every 100 training epochs) with every synthetic dataset.

In particular, we compared the effect of a linear AE, versus a non-linear AE in terms of latent manifold metrics. As the VAF of the linear AE was expected to be inferior to that of its non-linear counterparts (Portnova-Fahreeva et al., 2020), we aimed at obtaining a visual cue on why non-linear models are able to outperform a linear one in terms of reconstruction power.

Videos showing the concurrent evolution of the latent manifold and the reconstructed signals of the synthetic datasets with the different AEs are included in the Supplementary Materials.

2.3 Results

2.3.1 Study I:

Reconstruction power of different autoencoder networks

Figure 2.4 shows the synthetic datasets reconstructed by each of five AEs. The AE with a linear activation function failed to preserve the 3D structure of each dataset (Fig. 2.4, 2nd col). Similarly, an under-parametrized selu AE lost much of the input variance after DR (Fig. 2.4, 1st col), although the information related to the third dimension was not completely lost as in the linear case. When increasing the number of parameters, the nonlinear AEs were able to account for over 99% of the variance of all datasets. Interestingly, the type of nonlinear activation function did not impact the performance of the AE in terms of VAF (Fig. 2.4, 3rd to 5th col). The swiss-roll appears to be the most complicated dataset to reconstruct, as its VAF values were marginally but consistently lower than those of the fishbowl and the hemisphere (Table 2.1). However, each of the tested equally-parameterized nonlinear AEs accounted for over 98% of the swiss-roll variance. Finally, different initializations yielded comparable values of VAF, with standard deviations (Table 2.1) that were negligible for every dataset although consistently larger for the swiss-roll.

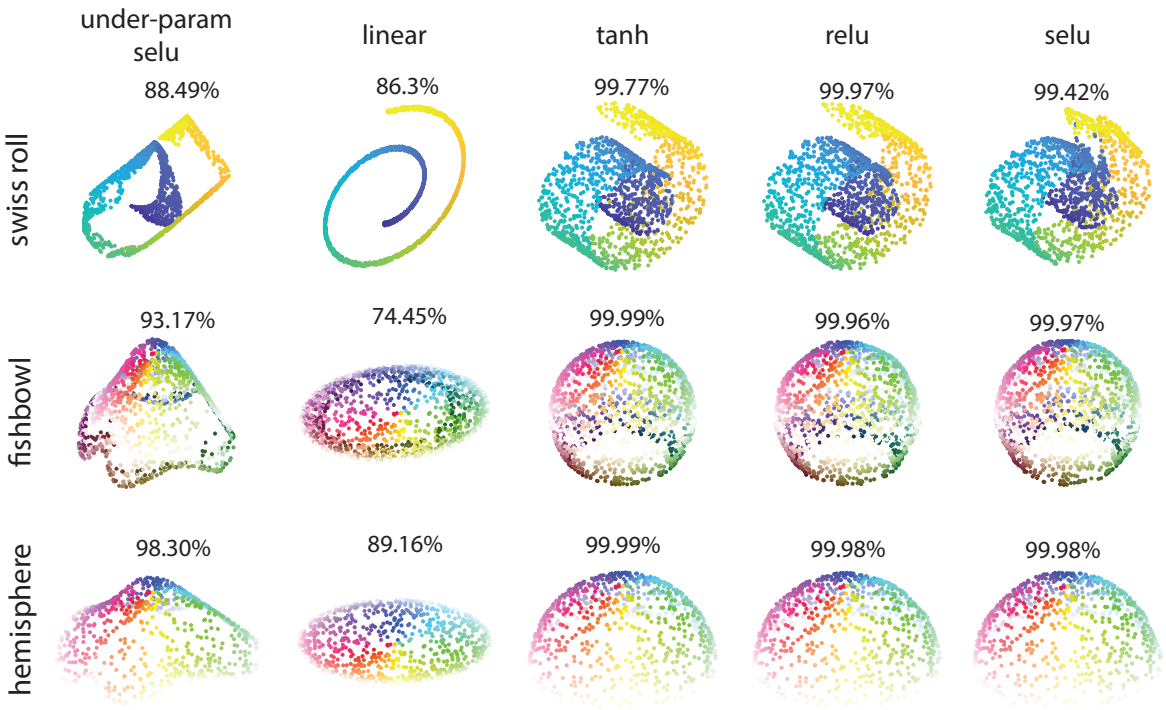

Figure 2.4 Synthetic data set reconstructed with each of five autoencoders. Each panel represents the reconstruction of the synthetic dataset in the corresponding row by the autoencoder in the corresponding column. Numbers above each reconstruction indicate Variance Accounted For (VAF).

Table 2.1 VAF (mean and std) of the reconstructed datasets for nonlinear AEs with tanh, ReLU, and selu activation functions. VAF is reported for both the training and test datasets for each synthetic dataset. Mean and std computed over trained AEs initialized with 20 different seeds.

	tanh VAF(%)		**ReLU VAF(%)**		**selu VAF(%)**	
	train	**test**	**train**	**test**	**train**	**test**
swiss roll	99.69 ± 0.21	98.77 ± 0.77	99.7 ± 0.26	99.18 ± 0.63	98.73 ± 0.88	98.04 ± 1.45
fishbowl	99.99 ± 0.006	99.99 ± 0.006	99.97 ± 0.01	99.94 ± 0.08	99.95 ± 0.01	99.94 ± 0.02
hemisphere	99.99 ± 0.04	99.98 ± 0.005	99.98 ± 0.004	99.97 ± 0.005	99.95 ± 0.01	99.94 ± 0.01

Corrected and uncorrected latent space of different autoencoder structures

Figure 2.5 presents the latent spaces of the synthetic datasets derived from the various linear and non-linear AEs without the cartographic correction. The shape of the latent space of each input appeared to be dependent on the structure of the AE used during training. It is interesting to note that while the VAF values and the reconstructed signals for the equally-parametrized non-linear AEs were fairly similar, their latent spaces appeared different. At a first glance, while the uncorrected latent spaces of the hemisphere and the fishbowl seem to be quite similar (Fig. 2.5, 2nd and 3rd row), those of the swiss roll substantially varied depending on the number of parameters of the AE (Fig. 2.5, 1st row, 1st col) and the type of activation function (Fig. 2.5, 1st row, 2nd to 5th col). Figure 2.6 shows the same latent spaces but after the cartographic correction. Figure 2.7 shows the corrected encodings on top of the corresponding cartogram. For each synthetic dataset, the cartographic correction allowed to represent the latent spaces with a metrics that is more faithful to that of the original signal. For instance, the corrected latent space of the hemisphere was much closer to the shape of a circle (Fig. 2.6, 3rd row) than its uncorrected counterpart. Similarly, the shape of the swiss roll corrected latent space appears to better preserve the curvature of the original swiss roll (Fig. 2.6, 1st row), with respect to that depicted in the uncorrected latent space. Figure 2.7 highlights how this metrics improvement was due to the distortions and warping introduced in the cartogram. Interestingly, the corrected latent spaces of the fishbowl (Fig. 2.6, 2nd row) appear quite different from its original uncorrected version. As shown in Fig.2.7 (2nd row), its cartogram seems to be folded onto itself, with the result that the encodings that were originally occupying the inner region of the latent space are now brought into the outermost region of the latent space, and vice versa.

2.3.2 Study II

Distance between aligned latent spaces before and after cartographic correction

The cartographic correction reduces the performance difference between autoencoders with different activation functions. Figure 2.8 shows the effect that different nonlinear activation functions have on the KLD between aligned latent manifolds for each synthetic dataset. For instance, the cell of the top-right corner of each panel shows the after- alignment KLD between the latent manifold for an equally-parametrized AE using tanh and the latent manifold for an equally-parametrized AE using selub. The KLDs between corrected latent manifolds were consistently lower than those between uncorrected latent manifolds. Figure 2.9 shows that the alignment was facilitated by the cartographic correction regardless of

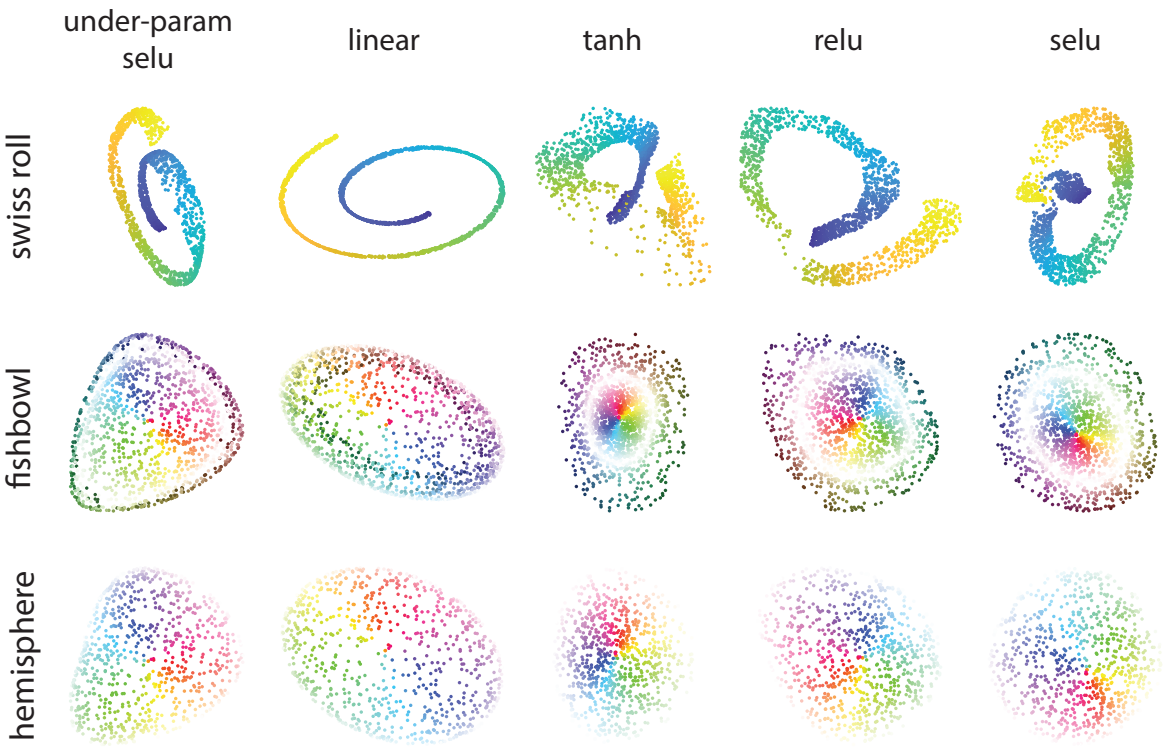

Figure 2.5 Figure 5: Uncorrected latent manifold for each synthetic dataset. Each panel represents the uncorrected latent manifold of the synthetic dataset in the corresponding row obtained with the autoencoder in the corresponding column. The scales of the latent manifolds have been normalized to a common size.

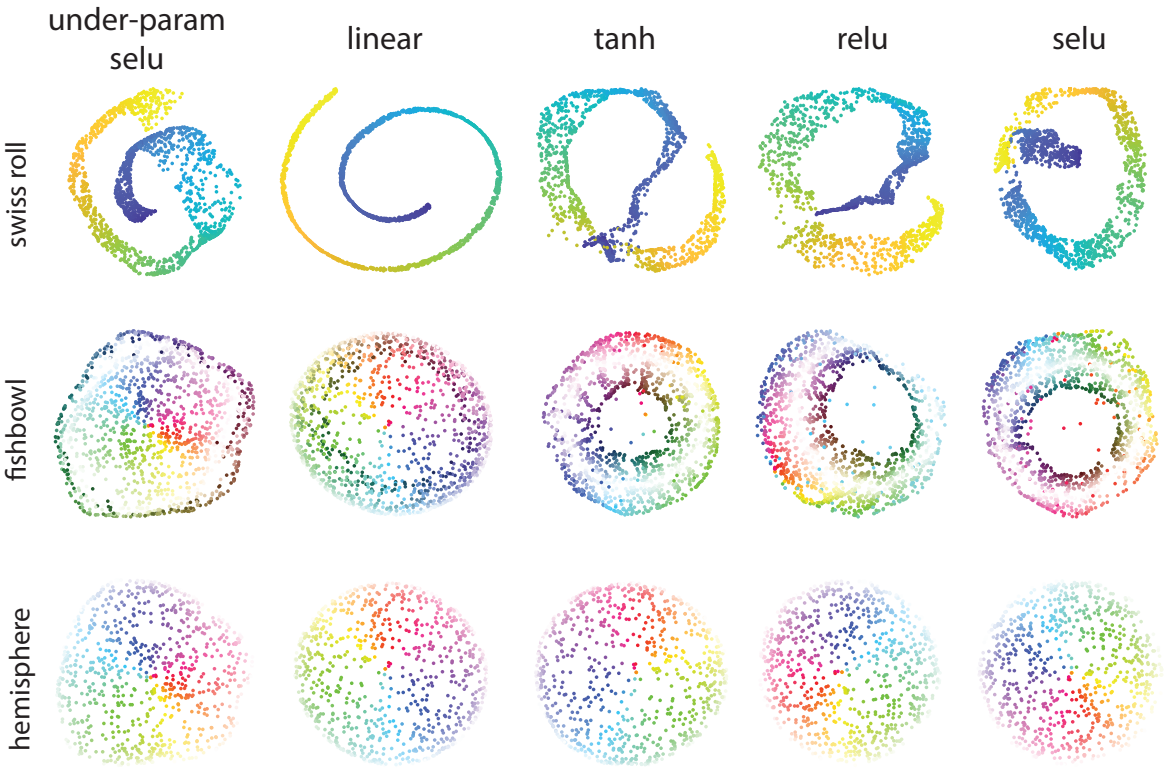

Figure 2.6 Figure 6: Latent manifold for each synthetic dataset after correction by the cartographic approach. Each panel represents the corrected latent manifold of the synthetic dataset in the corresponding row obtained with the autoencoder in the corresponding column. The scales of the latent manifolds have been normalized to a common size.

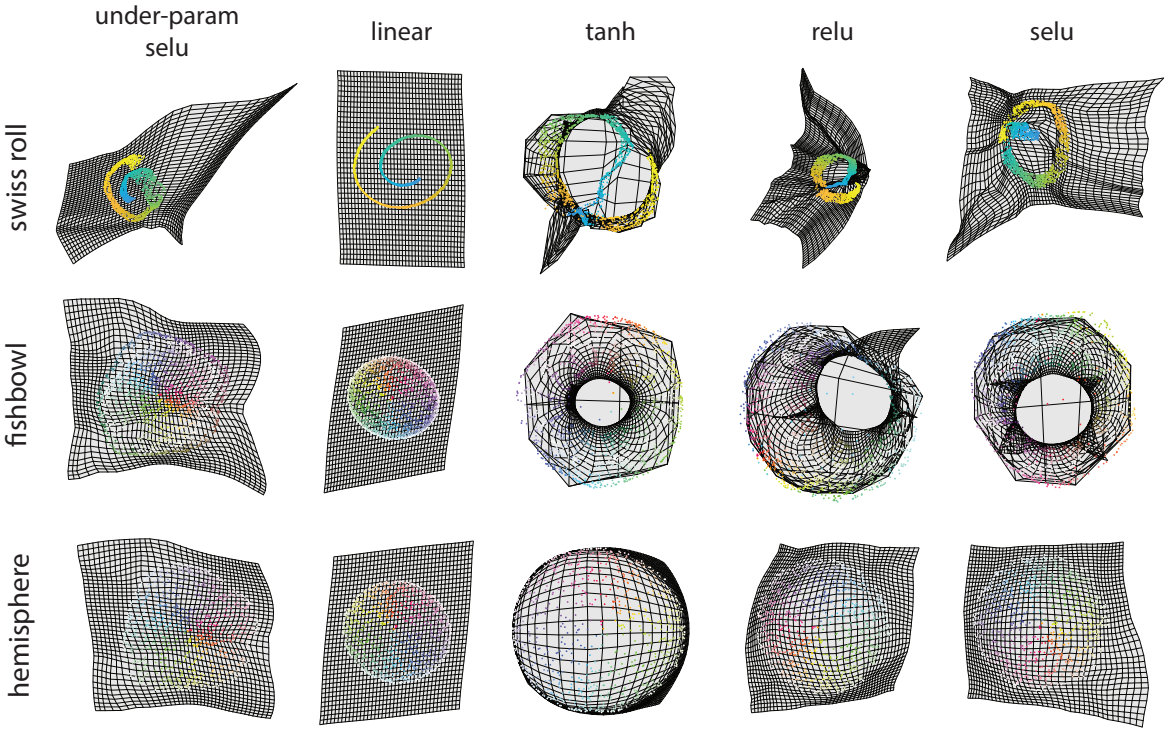

Figure 2.7 Figure 7: Cartograms obtained by correcting the grid in the latent space of each trained AE. Each panel represents the cartogram for the synthetic dataset in the corresponding row and for the latent space obtained with the autoencoder in the corresponding column.

Table 2.2 KLD (mean and std) between corrected (C) and uncorrected (UC) latent manifolds obtained after training nonlinear AEs with tanh, ReLU, and selu activation functions on each synthetic dataset. Mean and std computed over trained AEs initialized with 20 different seeds.

	tanh			**ReLU**			**selu**		
	C	**UC**	**p-value**	**C**	**UC**	**p-value**	**C**	**UC**	**p-value**
swiss roll	0.11 ± 0.1	0.1 ± 0.08	0.97	0.06 ± 0.06	0.1 ± 0.06	**0.005**	0.09 ± 0.04	0.31 ± 0.17	$<$ **0.001**
fishbowl	0.03 ± 0.02	0.11 ± 0.07	$<$ **0.001**	0.14 ± 0.11	0.04 ± 0.05	$<$ **0.001**	0.05 ± 0.04	0.06 ± 0.04	0.28
hemisphere	0.003 ± 0.003	0.08 ± 0.07	$<$ **0.001**	0.006 ± 0.005	0.05 ± 0.04	$<$ **0.001**	0.008 ± 0.007	0.15 ± 0.14	$<$ **0.001**

the activation function used during training, as the cartographic correction incorporated the metrics of the original dataset within each corresponding latent manifold.

To study the effect that different initializations had on the KLD between aligned latent manifolds obtained with each of the nonlinear activation functions, we ran a Wilcoxon signed rank for significant differences in the KLD between pairs of corrected (C) and uncorrected (UC) latent manifolds (values are shown in Table 2.2). The cartographic correction typically reduced the distance between latent manifolds for AEs trained from different initial conditions. However, the latent manifolds for the fishbowl dataset obtained from an AE trained with a ReLU activation function were more similar when uncorrected. This might be due to the fact that corrected latent manifolds derived from two different seeds have a thin outer layer at different locations within the latent manifold (Fig. 2.10, blue points for seed 0 and green points for seed 9).

2.3.3 Study III: Evolution of non-linearities in the latent manifold

We monitored the evolution of the corrected latent manifold, overlaid on the corresponding cartogram, derived from training a linear and a nonlinear (equally-parametrized and using the selu activation function) AE on the swiss roll dataset. Snapshots are shown in Figure 2.11. Note that in the early epochs of training, up to 100, the linear AE outperformed the nonlinear AE in terms of VAF, while the distortions captured by the corresponding cartograms were

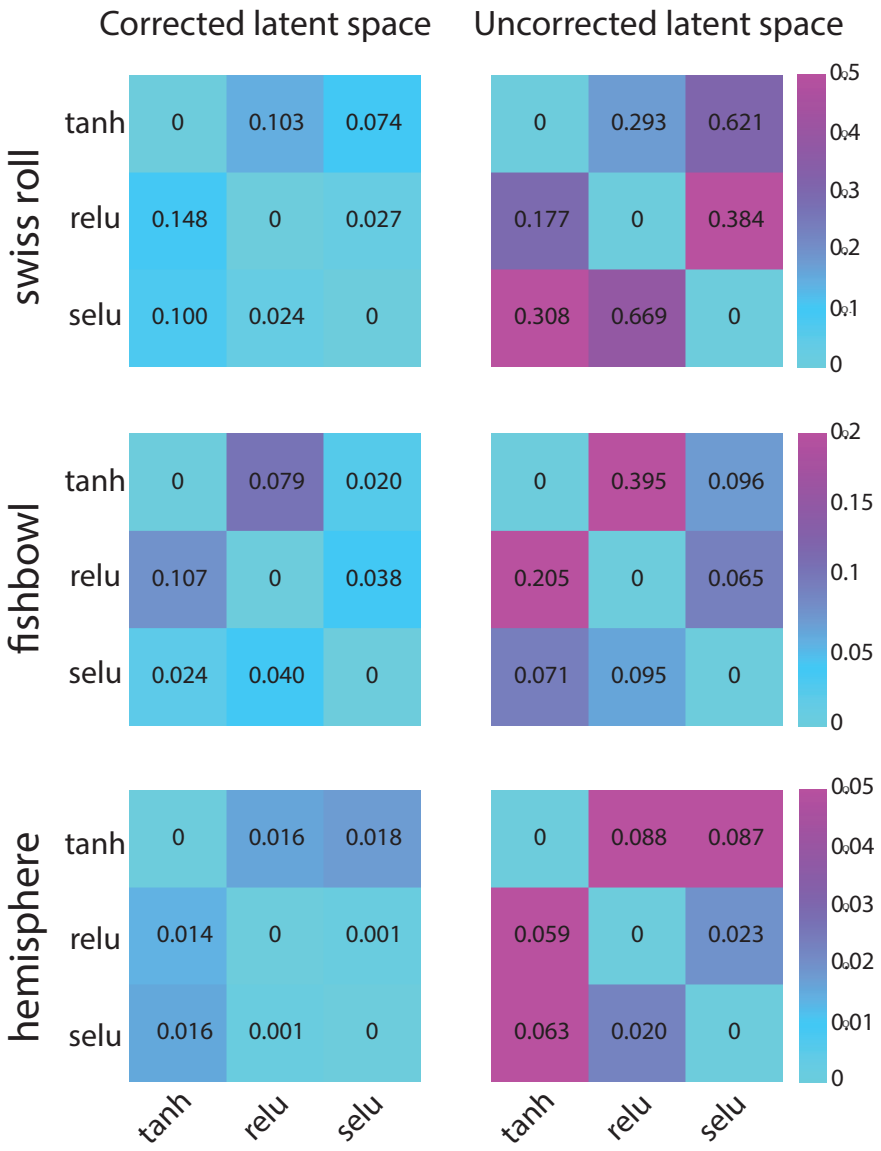

Figure 2.8 KLD values of corrected (left) and uncorrected (right) latent spaces for each pair of trained autoencoders with seed = 0.

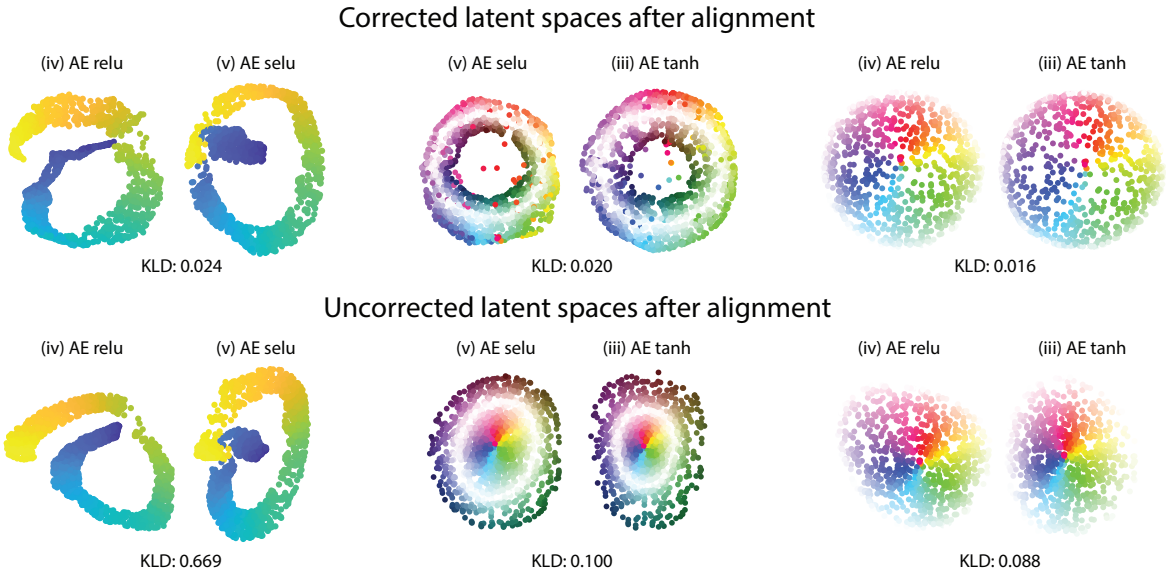

Figure 2.9 Example of two latent spaces trained with different activation functions and aligned after (top row) and before (bottom row) the cartographic correction for swiss roll (left), fishbowl (center) and hemisphere (right).

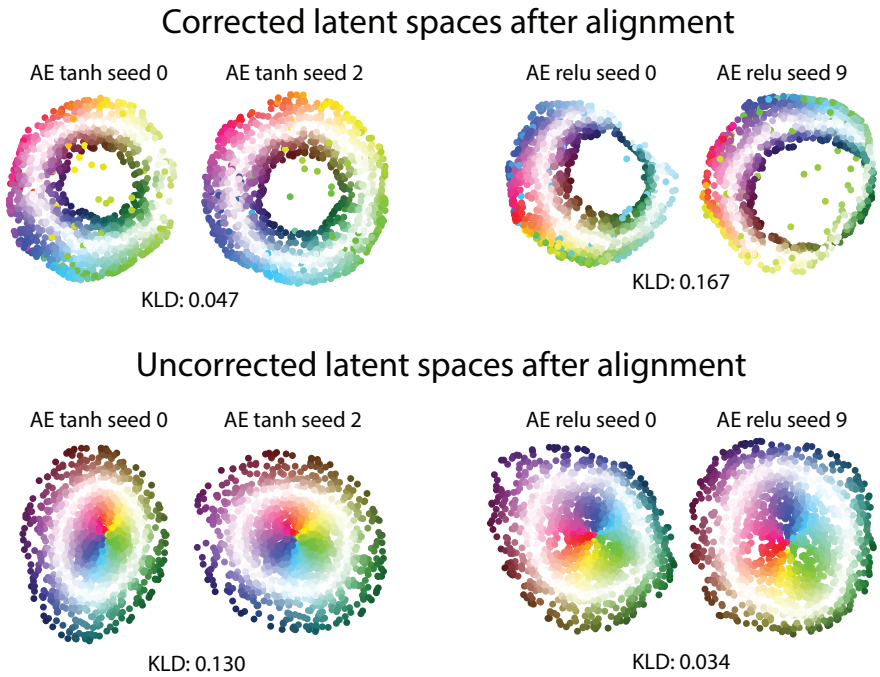

Figure 2.10 Example of two latent spaces trained with different seeds and aligned after (top row) and before (bottom row) the cartographic correction for fishbowl with tanh (left) and ReLU (right).

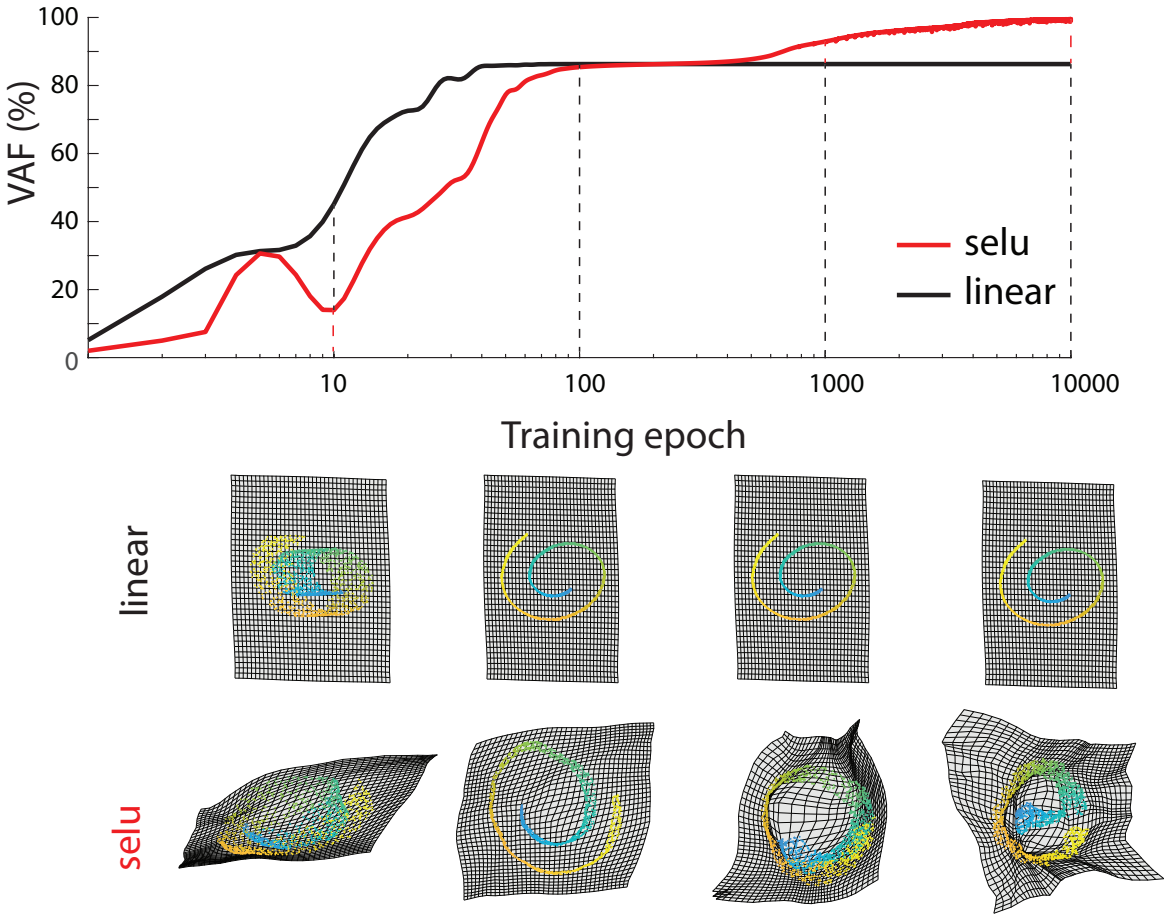

Figure 2.11 Top row: VAF values over training epochs for an AE trained with a linear (black) and a non-linear (red, selu) activation function. Note that the x-axis is plotted with a logarithmic scale. Vertical dashed lines indicate VAF values at 10, 100, 1000 and 10000 training epochs for both models. Snapshots of the corrected latent spaces, overlaid on the corresponding cartogram, for epochs 10, 100, 1000 and 10000 are shown in the second (linear) and third (selu) row.

quite similar. As training progressed beyond 100 epochs, the cartogram for the nonlinear AE became increasingly warped and its VAF continued to rise, while the cartogram and VAF of the linear AE remained unchanged. Around epoch 5000, the cartogram for the nonlinear AE started fluctuating around a stable structure while the VAF converged to a value close to 100%. The complete videos showing the concurrent evolution of the corrected latent manifold and the reconstructed datasets for the various linear and nonlinear AEs trained with each of the three synthetic datasets are included in the Supplementary Material.

2.4 Discussion

In this work, we proposed a rigorous approach based on MDS for finding a consistent latent representation of a dataset, independent of the parameters of the AE network trained to extract the data manifold. We showed that a combined use of AEs and MDS recovered the manifold of the input data from its latent encoding while providing the means to depict the nonlinearities embedded in its manifold during training. This gave us an intuition on why a nonlinear AE is able to outperform linear dimensionality reduction: when trained to reduce the dimensionality of a dataset with minimum loss, an AE with nonlinear activation functions can better match the local nonlinearities in the data. The cartographic corrections allowed us to approximate a universal latent manifold, independent of the details of the AE used to extract it. The proposed algorithm thus serves as a platform that can used to accurately visualize latent manifolds derived from high-dimensional data of different nature. This approach could have a powerful application in the context of human machine interaction that exploit a nonlinear latent manifold for controlling external devices through brain (Carmena et al., 2003) or body (Casadio et al., 2012) machine interfaces. The ease of use of such interfaces would benefit from a representation of the user's motor commands in a coordinate system that is metric-wise consistent with that where the user moves.

2.4.1 Is the manifold found by a trained AE unique?

When trying to estimate the latent manifold underlying a high-dimensional dataset, it would be reassuring if the solution was independent of the AE architecture and the training parameters used to obtain the trained AE that identifies the latent manifold. However, when training nonlinear AE networks, the landscape determined by the cost function is not convex and allows for multiple minima. Liao and Poggio demonstrated that over-parametrized deep networks may converge to one out of various solutions a flat global minimum, pertaining to one of many possible basins of attraction in the landscape of the cost function (Liao and Poggio, 2017). This implies that different initializations or weights perturbations will likely result in different convergence paths leading to different solutions that are almost equivalent in performance, as specified by the value of the cost function at the solution. Consistent with this description, in our simulations we observed only very small variations in the reconstruction error, or equivalently the VAF, of AEs trained with different initial conditions.

We also found that the VAF is invariant over the nonlinear activation function used in the AE network. The overall landscape of the cost function used to train the AE cannot

be assumed to be invariant with respect to the AE activation function. However, the VAF differences between AEs trained with different activation functions were also very small, implying that the solutions found by the various trained AEs were similar in terms the value of the cost function at the minimum.

It is however important to note that the fact that these different trained AEs find solutions with very similar performance does not imply that these solutions correspond to similar manifold encodings. In fact, our results show that the various uncorrected latent manifolds that result from training different AEs on a given dataset are not mutually consistent with respect to their shape, as they present different curvature and scaling. This is due to the fact that the projections of the original data points onto the uncorrected latent manifold do not preserve the metric relations they had in the original high-dimensional space. For instance, the curvature of the swiss-roll is not properly maintained in the latent manifold of either the linear (Fig. 2.5, 1st row , 2nd col) or the nonlinear AEs (Fig. 2.5, 1st row, 3rd to 5th col), although each nonlinear AE accounts for over 99% of the variance of this dataset. Similarly, the latent projections of the fishbowl and the hemisphere datasets depict a more ellipsoidal geometry than that of the original signals (Fig. 2.5, 2nd and 3rd row). The local nonlinearities in the data make it difficult to obtain a cartesian description of the manifold, an issue that might bring undesired consequences if we were to use the uncorrected latent manifold to highlight properties such as identifying where data points of different characteristics lie in the latent manifold.

We showed how to overcome this problem by reconciling the metrics of the latent manifold with the data metrics through the application of a cartographic correction to the latent manifold identified via an AE (Figs. 2.8-2.9, Table 2.2). The MDS used to obtain the cartograms allows the corrected latent manifold to preserve the metrics of the dataset. The resulting corrected latent manifolds are far more consistent across different AEs trained on a given dataset. As such, our approach is a step towards the identification of a unique manifold whose various instances are characterized by high statistical similarity (*i.e.*, low KLD) among themselves; they thus provide a more faithful representation of the dataset.

2.4.2 Linear vs nonlinear dimensionality reduction

The answer to the question of whether a nonlinear DR model is more appropriate than a linear one depends on the context. If we consider the problem of finding an accurate estimate of a latent manifold, two opposite viewpoints arise in the literature. While the majority of studies agree on the superiority of nonlinear models in capturing the variance of the

datasets (Portnova-Fahreeva et al., 2020; Tenenbaum et al., 2000), others claim that a linear DR approach yields more reliable manifolds than its nonlinear counterparts (Patel et al., 2015; Yan et al., 2020). This study shows that nonlinear DR methods clearly outperform linear ones when estimating latent manifolds. For a given same objective function (*i.e.*, maximizing the VAF), a linear approach (*e.g.*, PCA, linear AE) cannot yield better results an appropriately trained nonlinear one (*e.g.*, non-linear AE). Training that is stopped before proper convergence is reached might lead to a mistaken conclusion about the superiority of the linear model, such as when training only until the 100th epoch in Figure 2.11. When a sufficiently complex (*e.g.*, equally-parametrized) nonlinear AE is properly trained to convergence, it always yields a higher VAF than a linear AE (see Figs. 2.4 and 2.11), the latter being equivalent to PCA in terms of its reconstruction power (Plaut, 2018).

For the datasets considered in this study, even the simplest nonlinear AE (Fig. 2.4, 1st col) that introduced nonlinearities within a shallow architecture was able to consistently outperform the linear one. As expected, the values of VAF for trained linear or under-parametrized nonlinear AEs were significantly smaller than those for sufficiently parametrized nonlinear models. We claim that this performance deficit was due to these AEs not having enough power to warp their latent manifold to account for the curvature of the data manifold (see Fig. 2.7, 1st and 2nd columns). Our results highlight that incomplete training would yield to similarly poor performance in nonlinear and linear AEs, as latent manifold warping develops later on in the training. We also note that since the intrinsic dimensionality of the datasets used here is known to be 2, only sufficiently parametrized nonlinear AEs provided the correct estimate of intrinsic dimensionality based on the achieved VAF.

We emphasize that there are caveats to be considered before declaring nonlinear AEs the preferable choice. For starters, the architecture of a sufficiently parametrized nonlinear AEs is difficult to determine a priori, and can rapidly get quite complicated as the number of hidden layers/neurons is increased. The training procedure of such deep models requires great computational resources, an aspect of crucial importance in certain fields where a fast retraining of the model might be necessary to track non stationary data, as in the case of designing adaptive human-machine-interfaces ((De Santis and Mussa-Ivaldi, 2020). In addition, deeper networks are more prone to get stuck in local minima (Hinton and Salakhutdinov, 2006), thus requiring state-of-the-art optimization techniques to achieve convergent training.

In summary, it is the application context that should guide the choice between the simplicity of linear models or the customizability and power of nonlinear ones.

2.4.3 Cartograms of spherical surfaces

Among all the cartograms shown in the videos and summarized in Figure 2.7, those for the fishbowl data present a peculiarity: these cartograms seem to fold onto themselves. This condition might be explained by the Gauss's Theorema Egregium (Pressley, 2010), which states that curved surfaces can be bent without distortion/stretching if their Gaussian curvature is maintained. While a swiss-roll is a developable surface with zero curvature and a geometry that is isometric to a convex domain of Euclidean space (de Silva and Tenenbaum, 2003), a fishbowl has an intrinsic positive Gaussian curvature. Since a cartogram is a flat representation with zero curvature, an object with nonzero curvature cannot be represented by a cartogram without introducing distortions. Cartographers deal with this problem by constructing different projections of the Earth's surface that preserve different features such as area or shape. Here we have looked for a mapping that preserves the distances on a spherical surface when projected on a plane, a problem whose solution is not unique, ultimately leading to cartograms of spherical surfaces that are self-folding. This condition might be exploited when applying cartograms to real-world data of unknown structure, as it would highlight the presence of a spherical geometry intrinsic to the data.

2.4.4 Application of cartograms to human-machine interfaces

The existence of a latent manifold that encompasses signals derived from the brain (Gallego et al., 2017) or body (Casadio et al., 2012) is an important component in the design of human-machine interfaces. Both brain (BMIs) (Carmena et al., 2003) and body (BoMIs) (Farshchiansadegh et al., 2014) machine interfaces rely on the identification of a low dimensional latent manifold for extracting commands to control an external device. The process of obtaining the control signals from within the latent manifold is pivotal to ease the interaction between the user and the machine. Typically, the latent manifold is identified via unsupervised algorithms, while the mapping between the latent manifold and the coordinate system associated to the control output is selected in a supervised manner. For instance, BoMIs identify a low-dimensional subspace of movement synergies via Principal Component Analysis. Each principal component (PC) is then assigned to a specific control direction; for instance, in the case of controlling a computer cursor the 1st PC is mapped on the x-axis and the 2nd PC on the y-axis. This approach is mathematically consistent, as PCs are orthonormal as are the planar cartesian coordinates.

Linear models for dimensionality reduction have the advantage of providing an interpretable representation at a relatively low computational cost. The drawback is that they may

overestimate the dimensionality of the dataset, be it motor primitives or neural modes, when such primitives lie in a nonlinear manifold (Tenenbaum et al., 2000). This might increase interface complexity, an undesirable effect of needing a higher number of control signals. An inefficient approximation of the manifold would lead to a poor estimate of the user's motion intention and likely require frequent interface recalibration. Moreover, depending on the relationship between the input manifold curvature and the projection hyperplane, not all points in the manifold may be adequately mapped to the output as the two spaces will not be topologically equivalent.

As suggested by previous work, nonlinear dimensionality reduction methods provide a natural extension of linear methods to problems that require capturing a curved surface spanned by the dataset, as in the case of encoding hand kinematics (Portnova-Fahreeva et al., 2020); in such cases, the nonlinear better preserves the data topology within the latent manifold. However, introducing nonlinearities in a space of signals used for control gives up a major advantage of linear control: the superposition principle, the possibility to obtain a full repertoire of actions from the direct summation of simpler actions. Moreover, without properly accounting for distortions, a nonlinear model would generate a latent manifold that does not preserve the data metrics (*i.e.*, an uncorrected latent manifold). This scenario can create an inconsistency while operating an external device; for example, the extent and smoothness of a specific movement to be executed might not be reflected in the corresponding command to the interface device. These issues could result in non-intuitive and less user-friendly interfaces.

Nonlinear interfaces are fairly recent, and the few applications present in the literature do not address these potential problems (Pierella et al., 2018; Vujaklija et al., 2018). The cartographic correction proposed here used in conjunction with the interface control scheme would restore the consistency between data and latent manifold metrics, thus resulting in a control space that operates in a coordinate system metric-wise consistent with that space where the user moves, improving the usability of the interface. In addition, cartograms allow to quantify a distortion metric (DM) for latent manifolds, such as the ratio between the area of each square of the cartogram (defined in the corrected latent manifold) and that of the original rectilinear grid (defined in the uncorrected latent manifold). Such a metric would be particularly helpful in the case of latent manifolds for data whose geometry is not known a-priori, as is the case for the high-dimensional kinematic and neural signals commonly used in BMIs and BoMIs. Such distortion metric could highlight regions of the latent manifold are more linear (DM$\approx$1) and distinguish them from regions which are more nonlinear (DM$\neq$1). Such information could facilitate the choice between using a linear

or a nonlinear DR technique for obtaining the control manifold. Where linear regions are detected, one might prefer a linear method to take advantage of the superposition principle.

If the scenario requires the use of nonlinear DR, the choice of hyperparameters used to design and train the AE is nontrivial. One might prefer the AE with best reconstruction power (higher VAF). However, our results suggest that the shape of the resulting latent manifold should also be considered, especially when the latent manifold is intended to be used as the control space for an external device. For the three datasets studies here, the choice of the most suitable AE included the tuning of several hyperparameters. This tuning was not only driven by VAF maximization; for instance when comparing the effects of using different activation functions, differences in VAF were a mere 1%. In contrast with this apparently insignificant difference in VAF, visual inspection reveals notable differences in the geometrical shape the resulting latent manifolds; these might lead to impactful consequences during the online operation an interface that uses points in the latent manifold as input. We argue that the structure the resulting latent manifold should be considered when selecting the most appropriate DR approach for a given scenario.

The use of the cartographic correction proposed and developed here in conjunction with an interface for control would facilitate the choice of DR approach, as it would make the latent manifolds derived with different AE hyperparameters as consistent as possible. The corrected manifolds incorporate the correct dataset metrics, thus specifying the geometry of the admissible set of device configurations.

Part II

Linear and nonlinear dimensionality reduction for unsupervised training and adaptation for body machine interfaces

Preface

After analyzing the properties of both linear (PCA) and nonlinear (AE) dimensionality reduction (DR) techniques, I focused on their applications within the control-scheme of a body-machine interface (BoMI) (Casadio et al., 2012). In their first implementations, BoMIs (Farshchiansadegh et al., 2014) relied on the identification of a low-dimensional latent space using a linear model, such as PCA, that defined a forward map from which to extract commands for controlling an external device. Linear control gives the possibility to obtain a full repertoire of actions from the direct summation of simpler actions. Also, linear models have the advantage of providing an interpretable and mathematically unambiguous lower-dimensional representation at a low computational cost. In Chapter 3, I present a novel BoMI that could be used for therapeutic purposes. Specifically, I designed a BoMI by applying PCA to kinematic and muscular signals recorded from the upper body and combined them into a hybrid signal to provide users with the possibility to switch seamlessly between movement and EMG control. This allows using the BoMI as a tool for promoting the engagement of selected muscles while overcoming the limitations of an EMG-only control (due to EMG having a poorer signal-to-noise ratio compared to kinematics).

As discussed in Part I of this thesis, the drawback of using linear DR methods is that they tend to overestimate the dimensionality of the motor primitives, thus risking to detriment the interface usability. Non-linear DR models may provide a natural extension of linear methods to problems where capturing a curved surface spanned by the input is required. However, there has been only preliminary evidence regarding the applicability of non-linear AEs to the control of low-dimensional devices (Pierella et al., 2018; Vujaklija et al., 2018). Hence, in this project I aimed at validating the use of nonlinear AEs in BoMI applications by controlling both 2D (computer cursor, Chapter 4) and 4D (robotic manipulator, Chapter 5) devices.

In previous iterations of BoMIs, the forward map used to derive commands for the external device remained static over the course of the training after an initial offline tuning (*i.e.*, calibration phase). Therefore, the user is left with the burden of learning how to

use the interface for achieving new goals. In Chapter 4, I propose an innovative adaptive interface that makes use of a nonlinear AE trained iteratively to perform online manifold identification and tracking, with the dual goal of reducing the need for interface recalibration and enhancing human-machine joint performance. Importantly, the method can be applied in the earliest stages of interface operation, when the formation of new neural strategies is still on-going (Oby et al., 2019). Moreover, the proposed approach does not require interrupting the operation of the device and it neither relies on information about the state of the task, nor on the existence of a stable neural or movement manifold, in contrast with other adaptive interfaces described in the literature (Degenhart et al., 2020; DiGiovanna et al., 2008; Mahmoudi et al., 2008; Sanchez et al., 2009).

Chapter 3

A hybrid Body-Machine Interface integrating signals from muscles and motions

3.1 Introduction

One of the many consequences of spinal cord injury (SCI), stroke and other neurological conditions is the loss of voluntary control of muscles. Impaired voluntary control often manifests with muscle weakness and increase of undesired co-activations of muscles during voluntary contractions (Adams and Hicks, 2005; Cremoux et al., 2016; Watkins et al., 2002). The development of such dysfunctional activation patterns affects the ability to perform selective movements and coordinate joint motion, resulting in low performance during functional tasks. Thus, decreasing those abnormal muscle activations, as well as restoring control to neuromuscular regions that have lost cortical input but remain in the zone of partial preservation, become of primary importance. Body-Machine Interfaces (BoMIs) (Casadio et al., 2012; Mussa-Ivaldi et al., 2011) could be valuable instruments to reach this goal, combining assistive and rehabilitative purpose.

BoMIs convert body-derived signals (motion, forces, or neurophysiological activities) into control signals, which in turn are used to operate external devices. As a result, they

The content of this chapter has been published as Rizzoglio, F., Pierella, C., De Santis, D., Mussa-Ivaldi, F.A. and Casadio, M., 2020. A hybrid body-machine interface integrating signals from muscles and motions. *Journal of Neural Engineering.*

enable individuals with disabilities to overcome some of their impairments. The primary use of the BoMI lies in it being an instrument to recover independence after paralysis, although its potential value can also be considered in other situations involving the control of a device by body motions (Miehlbradt et al., 2018) (as in teleoperation (Ajoudani et al., 2012)). The rationale for the use of body-signals is to maintain and support the engagement of available movements, which is a significant goal after SCI and other neurological conditions. Among the applications of a BoMI, earlier works by our group have focused on the use of motion signals derived from inertial measurement units (IMUs) (Abdollahi et al., 2017), (Pierella et al., 2017b; Seáñez-González et al., 2016; Summa et al., 2015; Thorp et al., 2015). Although useful for assistive purposes in individuals with impaired functions, IMU signals do not selectively target users' muscle activations. For this purpose, muscles must be explicitly integrated in the control architecture.

Myoelectric interfaces have been widely studied (Vujaklija et al., 2018), especially in the context of prosthetics (Fani et al., 2016; Zecca et al., 2002), but also for rehabilitative purposes (Mulas et al., 2005), both in SCI (Rizzoglio et al., 2019) and stroke (Wright et al., 2014). However, EMG signals are inherently noisier than motion signals. The poorer signal-to-noise ratio has an impact on control, leading to a less efficient operation of the external device. To mitigate the effects of neuromuscular noise, one possibility would be to integrate EMG and kinematic signals into a hybrid control system. The combination of IMU and EMG signals has been broadly investigated in the biomedical field. However, most studies have exploited the fusion of EMGs and IMUs for purposes that did not directly involve the use of muscles in the control scheme. IMU and EMG signals have been used together as a mean to improve classifiers for gesture (Georgi et al., 2015) and American Sign Language (Wu et al., 2016) recognition, as well as capturing the motion of the arm (Xu et al., 2016). When muscle activities were used for control, the design of a control map was based on supervised methods, as in the case of a hybrid-based classifier for the control of a computer cursor (Forbes, 2013; Xiong et al., 2011), a trans-humeral prosthesis (Lauretti et al., 2016), or a powered wheelchair (Kundu et al., 2018). This approach, however, limits the user's ability to operate the device in a continuous manner, as the actions to be performed are predefined and not fully tailored to the user's available motor abilities. Unlike supervised algorithms, the BoMI approach proposed the use of an unsupervised control algorithm such as Principal Component Analysis (PCA (Wold et al., 1987)). In this way, the interface can adapt to one's residual movements and/or muscle activations, potentially facilitating their continuous interaction with the interface.

In the context of BoMIs, the integration of kinematic- and muscle-based control had not been explored yet. In this study, we envisioned a hybrid Body-Machine Interface that would allow to seamlessly switch between kinematic and muscle control modalities. The goal is to provide a way for users to efficiently control the assistive device while maintaining and promoting engagement of selected muscles. In this way, the proposed algorithm would encourage the dual goal of providing the instrument for assistive support and therapeutic intervention.

We developed and tested an interface combining IMU and EMG signals into a hybrid control signal and validated its use for controlling a computer cursor. We tested and compared three different BoMI modalities, IMU-only, EMG-only, and hybrid—IMU and EMG combined, to assess which one would support a more intuitive and easier-to-use interface. Finally, we characterized the motor strategies followed by participants in the three different BoMI modalities and assessed the contribution of the IMU and EMG signals in the hybrid modality.

3.2 Methods

3.2.1 Experimental apparatus

This study focused on movements of upper arms and shoulders. Our purpose was to involve more proximal upper limb muscles that are commonly affected by stroke or cervical spinal cord injury and are often targeted by therapeutic interventions (Brucker and Buylaeva, 1996; Giuffrida and Crago, 2001; Harburn and Spaulding, 1986; Hincapie and Kirsch, 2007; Tropea et al., 2013; Wright et al., 2014). In addition, arms and shoulders are easily accessible and offer the opportunity for individuals to operate the interface from a seated position with minimal discomfort. The BoMI recorded eight muscular and eight motion signals bilaterally from the shoulders and upper arms and combined them in two control signals for a computer cursor.

To record motions of the upper body, we used four IMUs (3 Space Sensors, Yost Labs, Portsmouth, OH, USA) placed bilaterally on arms and shoulders of each participant as in figure 3.1. Each IMU sensor was calibrated to transform the raw data from tri-axial accelerometers and gyroscopes into sensor orientation in the Euler-angle format (pitch, roll, and yaw). The calibration was required to optimize the performance of the filters. Since the yaw measurements were affected by drift, we decided to record only the pitch and roll values, which remained stable for the duration of the experiments. To record the muscular activity,

we used a 16-channels wireless EMG system (Wave Plus, Cometa, Milan, Italy). Muscle activity was digitally acquired via a C# API at a sampling frequency of 2 KHz. We recorded the activity from eight muscles of the left and right arms and shoulders, in particular biceps brachii, lateral triceps, lateral deltoid, and upper trapezius. The raw EMG signals were first bandpass-filtered (2nd order IIR Butterworth, $fc = 30 - 450Hz$) to remove high-frequency noise and low-frequency artifacts caused by movement of the electrode relative to the skin. The signals were then rectified and low-pass-filtered (2nd order IIR Butterworth, fc = 2Hz) to obtain the EMGs envelope.

The interface software was developed in C# and based on a multi-threaded architecture to synchronize the acquisition of IMU and EMG signals. The interface consisted of two threads that handled continuous acquisition and online processing of IMU and EMG signals and a thread to handle cursor control and graphic for a reaching task. The graphical part of the BoMI was developed using OpenTK, a C# graphic library that provided access to graphic tools defined in OpenGL. The update of the graphic frame ran at 50 Hz.

In the BoMI scheme (Casadio et al., 2012), the mapping from body to control signals consisted of two steps: (1) the unsupervised identification of the latent space by dimensionality reduction methods (Hyvärinen and Oja, 2000; Kramer, 1991; Schölkopf et al., 1997), (2) the supervised selection of the coordinate system over the latent space based on user's preferences. The dimensionality-reduction method used in this study was PCA.

All participants performed a 60 s guided calibration, during which they were asked to replicate movements of arms and shoulders performed by the experimenter. These movements were performed one at a time (i.e. first shoulder flexion, then shoulder extension, then elbow flexion, etc). This ensured that each participant had a homogenous calibration data set inclusive of all possible joint movements. During the execution of a selected movement, movements of the other joints were not restrained. The instructions were to explore the range of motion without performing extreme or uncomfortable movements. In this phase, both EMG and IMU data were collected and used as a training dataset for the PCA algorithm to derive the control map. Two mappings were computed using kinematic and muscular calibration data independently, in order to control the two-dimensional cursor either with the eight IMU or the eight EMG signals. Each mapping was obtained by applying an eigenvalue decomposition separately of IMU's and EMG's calibration data covariance matrices. The resulting eigenvectors were ordered by the decreasing values of variance accounted for. Thus, the first two eigenvectors - $\boldsymbol{h_1}$ and $\boldsymbol{h_2}$, where $\boldsymbol{h_i^T} = [h_{i,1} \cdots h_{i,8}]$, i = 1 : 2 -identified the hyperplanes where the body signals had the greatest variance. The mapping from body-space

(IMU or EMG) - $\boldsymbol{q}$, to control-space (x-y coordinates of the cursor)- $\boldsymbol{p}$, followed the same structure as reported in (Casadio et al., 2010; Farshchiansadegh et al., 2014)

$$\begin{aligned} \boldsymbol{p}_{[2x1]} &= \boldsymbol{G}_{[2x2]} \cdot \boldsymbol{H}_{[2x8]} \cdot \boldsymbol{q}_{[8x1]} + \boldsymbol{p0}_{[2x1]} \\ \boldsymbol{p} &= \begin{bmatrix} \frac{d_1}{\sqrt{\lambda_1}} & \mathbf{0} \\ \mathbf{0} & \frac{d_2}{\sqrt{\lambda_2}} \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{h_1^T} \\ \boldsymbol{h_2^T} \end{bmatrix} \cdot \begin{bmatrix} q_{1,1} \\ \vdots \\ q_{8,1} \end{bmatrix} + \boldsymbol{p_0} \end{aligned} \tag{3.1}$$

To ensure full coverage of the workspace, we included an adjustement $\boldsymbol{G}$ to rescale $\boldsymbol{h_i}$ by the eigenvalues λ_i and the desired width d_1 and height d_2 of the workspace. Moreover, $\boldsymbol{G}$ could be modified to impose a customized rotation and scaling to the workspace to match participant's preference. The offset vector $\boldsymbol{p_0}$ was chosen to make the origin of the body motion space match a corresponding reference position of the cursor.

We implemented three different BoMI mapping modalities: IMU-only, EMG-only, and hybrid. The following equation defines the cursor position vector, $\boldsymbol{p}$, for all modalities:

$$\boldsymbol{p} = \alpha \boldsymbol{p}_{IMU} + (1-\alpha)\boldsymbol{p}_{EMG} \tag{3.2}$$

If $\alpha = 1$, the control was carried out by IMUs only (IMUmod), while if $\alpha = 0$, the control was carried out by EMG alone (EMGmod). Values of $0 < \alpha < 1$ implemented a weighted sum of the two modalities of input (HYBmod). Specifically, in our study, the hybrid control was defined with $\alpha = 0.5$ as we aimed to give equal weight to the original and predicted IMU values.

$\boldsymbol{p}_{IMU}$ was the vector containing the coordinates of the cursor obtained by applying PCA to the IMU signals ($\boldsymbol{q}_{IMU}$) as in 3.1. $\boldsymbol{p}_{EMG}$ was defined as a second-degree polynomial of $\boldsymbol{q}_{EMG}$:

$$\boldsymbol{p}_{EMG} = \boldsymbol{W}_{EMG} \cdot \boldsymbol{Q}_{EMG} \tag{3.3}$$

with

$$\boldsymbol{Q}_{EMG} = \begin{bmatrix} 1 \\ \boldsymbol{q}_{EMG} \\ diag(\boldsymbol{q}_{EMG}\boldsymbol{q}_{EMG}^{\boldsymbol{T}}) \end{bmatrix} \tag{3.4}$$

where $\boldsymbol{q}_{EMG}$ is the 8×1 vector of EMG envelopes. $\boldsymbol{W}_{EMG}$ is a 2×17 matrix which elements varied according to the BoMI modality, either EMGmod or HYBmod. In case of EMGmod it was defined as:

$$\boldsymbol{W}_{EMG} = \begin{bmatrix} 0 & \boldsymbol{h}^{\boldsymbol{T}}_{1_{EMG}} & 0 \\ 0 & \boldsymbol{h}^{\boldsymbol{T}}_{2_{EMG}} & 0 \end{bmatrix} \tag{3.5}$$

where $\boldsymbol{h}^{\boldsymbol{T}}_{i_{EMG}}$ were respectively the first and second eigenvectors obtained by applying PCA to $\boldsymbol{q}_{EMG}$. The zero vector $\mathbf{0}$ canceled out the quadratic terms $diag(\boldsymbol{q}_{EMG}\boldsymbol{q}^{\boldsymbol{T}}_{EMG})$ in $\boldsymbol{Q}_{EMG}$. Thus, in EMGmod, equation 3.3 reduces to equation 3.1 with $\boldsymbol{p} = \boldsymbol{p}_{EMG}$ representing the vector containing the coordinates of the cursor obtained by applying PCA to EMG ($\boldsymbol{q}_{EMG}$) signals.

For the hybrid modality (HYBmod) we hypothesized that a direct combination of IMU and EMG signals might result in an inefficient control for the user, due to the heterogeneous character of these signals. To encourage a smooth integration and transition between modalities, we decided to transform the EMG signal into an IMU-equivalent signal by using a regression-based approach. In earlier studies, this method has been employed to decode kinematic signals from surface EMG signals used in simultaneous and proportional myoelectric control of prostheses (Hahne et al., 2014; Krasoulis et al., 2015). We implemented a multivariate polynomial regressor, which predicted the cursor coordinates obtained by IMU signals, starting from EMG envelopes. We used a 2nd order polynomial regression to fit the nonlinear relationship between kinematic and muscular signals. Regression was performed on the calibration dataset: the first 30 s were used as training set while the last 30 s were used as test set. We chose to analytically solve the nonlinear regression problem by using the Normal Equation and a L_2 regularization with a least-squares cost function:

$$\boldsymbol{W}_{EMG} = (\boldsymbol{Q}^{\boldsymbol{T}}_{EMG}\boldsymbol{Q}_{EMG} + \lambda \boldsymbol{I})^{-1}\boldsymbol{Q}^{\boldsymbol{T}}_{EMG}\boldsymbol{p}_{IMU} \tag{3.6}$$

where $\boldsymbol{Q}_{EMG}$ was defined as in equation 3.4 and the value of the regularization parameter λ was validated via a search in the range $[0 : 10^{-3} : 1]$.

3.2.2 Participants

Fifteen unimpaired participants (eight females, age 23.7 ± 1.39 years) were enrolled in this study. They did not have any known history of neuromotor or musculoskeletal disorders and exhibited typical joint range of motion and muscle strength. The research was conducted at the University of Genoa in conformity with the ethical standards laid down in the 1964 Declaration of Helsinki for the protection of research subjects and all the study procedures were approved prior the beginning of the study by the local ethical committee (Comitato Etico Regionale—regione Liguria—N. Registro ASL 13/13). According to these procedures,

each participant signed a consent form for participating in the study and for the publication of the data in the deidentified form.

3.2.3 Experimental protocol

The protocol consisted of a reaching task with three different BoMI modalities:

- **IMUmod**–($\alpha = 1$) :the cursor was controlled only by IMU signals;
- **EMGmod**–($\alpha = 0$): the cursor was controlled only by EMG signals
- **HYBmod**–($\alpha = 0.5$): the control of the cursor was shared by IMU and EMG in equal proportion.

IMU and EMG signals were always recorded in every BoMI modality. Participants sit in front of a 15.6in LCD computer screen, positioned about 1 m away at eye level. The current position of the cursor and targets were displayed on this screen as circles of 0.3 cm and 0.8 cm diameter respectively. Participants were asked to move the cursor over the targets and to remain within each target area for 0.5 s. The reaching task was organized in a center-out, out-center fashion. A new peripheral target was not presented if the cursor was not in the central (HOME) target. This ensured that each center-out motion started from the central target. No time constraint was imposed for reaching the targets. The protocol included two phases of reaching: training and test.

Training phase

The training phase took place in six reaching epochs—two for each BoMI modality. The order of presentation of each BoMI modality was pseudorandomized across the six epochs for each participant, such that all three modalities would appear at least once every three epochs. An example of randomization is reported in figure 3.1. During each epoch, three targets (figure 3.1, grey targets) positioned at 6 cm from the center of the computer screen were presented in three different directions: $0°$, $30°$, and $-30°$. The three peripheral targets were presented twelve times in pseudorandom order, with the condition that each peripheral target was not presented again before all three targets had been reached. Therefore, in this phase participants performed a total of 72 center-out movements for each BoMI modality.

Test phase

The test phase consisted of three reaching epochs— one for each BoMI modality-with a random order per participant. The goal of this phase was to test if and to what extent participants were able to transfer the skills acquired during the training phase to conditions where they had to move toward (i) different directions and/or (ii) to different displacement amplitudes (scaling-expansion) with respect to the training phase as depicted in blue in figure 3.1 (blue - untrained and grey - trained targets). Each of the 11 targets was presented four times in pseudorandom order. As shown in figure 3.1, the targets were distributed radially at three distances from the HOME target (L/2: 3 cm, L: 6 cm and 3L/2: 9 cm) at an angle of $30 \pm^{\circ}$ and 0° (trained directions) and $\pm 15^{\circ}$ (untrained directions, only at distance L). In this phase participants performed a total of 44 center-out movements for each BoMI modality.

3.2.4 Performance measures

Three different parameters were chosen to evaluate participants' performance:

- **Movement Time** (**MT**): time to reach the peripheral targets after leaving the HOME target;
- **Linearity Index** (**LI**): maximum deviation from the straight line connecting the beginning and end of cursor movement divided by nominal distance. This is an index of straightness of cursor movements;
- **Movement Smoothness** (**MS**): trajectory smoothness was assessed by the number of peaks in the cursor speed profile. We considered every peak larger than a threshold that was set to be 15% of the maximum speed of each trajectory.

3.2.5 Contribution of IMU and EMG signals in HYB modality

Special focus was given to the hybrid BoMI modality. Since it was defined as the combination of two different signals—IMU and EMG—we studied how these contributed to the participants' performance in HYBmod. We performed two different types of analysis, one based on cursor position and the other on cursor velocity. We resampled each reaching trial between 0% (when participants exited the HOME target) and 100% (as soon as they entered the peripheral target).

- **Position analysis**: we considered the position of the cursor (p) during reaching and the two contributions to p coming from IMU and EMG signals (p_{IMU} and p_{EMG})

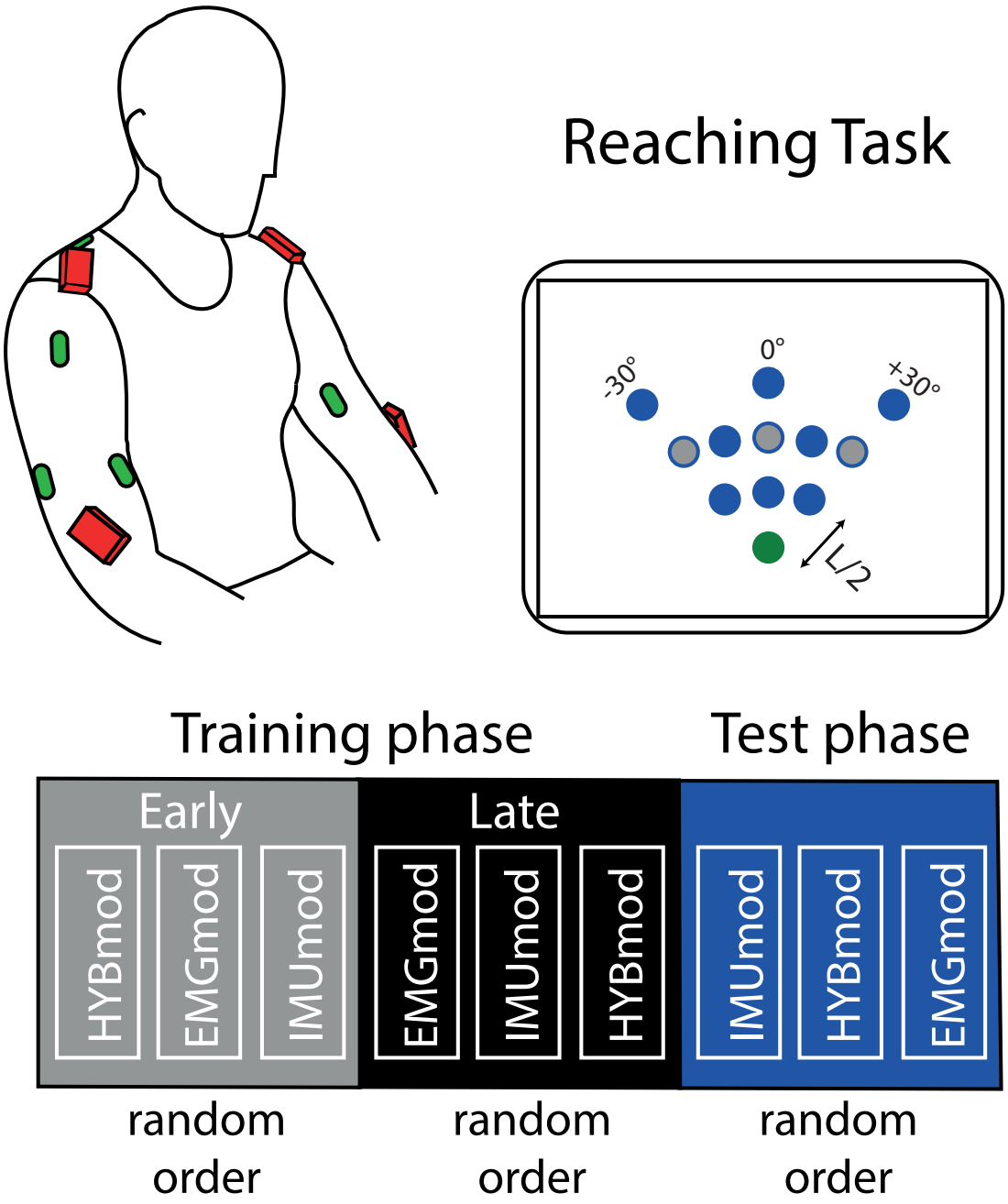

Figure 3.1 Setup for the reaching task and training protocol. The participant was sitting in front of a computer and was controlling a cursor using signals generated by IMUs (red boxes) and EMG electrodes (green circles). The EMG electrodes are labelled for the right side of the body as follows: biceps-BIC, triceps-TRIC, deltoid-DELT and trapezius-TRAP. Training (grey) and test (blue and grey) targets were presented in a 60° triangular-shape space. Three training targets were set in three different directions (±30°, 0°) at the same distance L of 6 cm from HOME (green) target. Eleven test targets were set in five different directions (30°, ±15°, 0°) at three different distances (L/2 = 3, L = 6, and 3L/2 = 9 cm) from HOME target. Thus, during the test phase, participants were asked to reach trained (grey) and untrained (blue) targets.

respectively, the latter after applying the regression map. First, we were interested in estimating how IMU and EMG contributed to move the cursor towards the target. We computed the angle between p_{IMU} and p_{EMG} with respect to the peripheral target throughout the entire reaching movement (*angle from target*). Then, we computed the norm of the error between the peripheral target and p_{IMU} and p_{EMG} to assess whether the two contributions decreased monotonically during the whole reaching movement. Finally, we computed the spectrum of the cursor position, p_{IMU} and p_{EMG}, to evaluate the contribution of the IMU and EMG components in the frequency domain.

- **Velocity analysis**: we computed the velocity vectors and their norm (speed) of the cursor v and of its two components v_{IMU} and v_{EMG}. Then we projected vIMU and vEMG onto v. The larger the projection, the stronger the contribution of the corresponding component to the overall velocity of the cursor (expressed as percentage of the overall velocity). We then computed the two angles between v_{IMU} and v and between v_{EMG} and v as an index of *trajectory similarity*.

3.2.6 Control strategies within each BoMI modality

Beside performance indices, we aimed to evaluate how participants controlled muscles and movements under each of the BoMI modalities. While the reaching task was inherently planar and two-dimensional, participants could reach each target with a theoretically infinite variety of body signal patterns. Here, by 'control strategy' we intend any particular solution to the reaching task adopted by a participant, that is, any inverse model of the BoMI forward map. Thus, a control strategy transformed a desired position of the cursor (*i.e.* a target) into a single pattern of body signals driving the cursor in each BoMI modality. We computed the **Variance Accounted For** (**VAF**), defined as the percentage of variance of a data set explained by each principal component (PC) during test epochs for each BoMI modality. Considering only two PCs, the higher the value, the more planar the data set. Thus, we used VAF also to evaluate IMU and EMG planarity. We consider planarity to be an important property of these signals: while these are driving the cursor in a planar motion, there is no constraint for them to be planar as well.

Additionally, we investigated how practicing the task with one modality influenced the IMU/EMG signals regardless of their direct participation to cursor control. Specifically, we considered 'control signal' the signal that was actively being mapped into cursor position (*e.g.* IMU during IMUmod), and 'auxiliary signal' the signal that was recorded but did not contribute to cursor control (*e.g.* EMG during IMUmod). As our goal was to evaluate how

much the control signal changed during training compared to the auxiliary signal, we had to devise a way to compare the variation of two physically incommensurable signals, one referring to the muscle activity, the other to body kinematics. We adopted as measure for this comparison the target information expressed by either IMU or EMG signal, which does not depend on the signal physical dimensions. Thus, we considered the 8D vectors of IMU or EMG signals at the instant when training targets were reached (*i.e.* **endpoint**). For each of those vectors we trained a classifier (SoftMax regression (Gao and Pavel, 2017)) to map the body signals, either the control or the auxiliary signals, to their corresponding target. The input of the classifier was a 36×8 matrix, where 36 are the targets reached in both the early and late training of each BoMI modality. Specifically, we trained a classifier on the late training epochs of each BoMI modality (IMUmod, EMGmod, and HYBmod) and tested it on the early training epochs of the same modality (see supplementary materials for more details (available online at stacks.iop.org/JNE/17/046004/mmedia)). Thus, a total of six classifiers were trained, one per each combination of signal and modality. We performed a 6-fold cross validation and computed the **Classification Accuracy** (**CA**) of each classifier, defined as:

$$\boldsymbol{CA} = \frac{\text{Number of correct classifications}}{\text{Total number of endpoints}}(\%) \tag{3.7}$$

High values of accuracy were expected if endpoints in early training were similar to those in late training (where the classifiers were trained on). We hypothesized that, since IMU and EMG are physiologically correlated signals, changes in one of them would correspond to changes in the other, regardless of which is the control, and which is the auxiliary signal.

3.2.7 Statistical analysis

To test the effect of time and BoMI modality on the indicators related to kinematic performance during training, we ran repeated measures ANOVA (rANOVA) with two within-subject factors: BoMI modality (1–3: IMUmod, EMGmod, HYBmod) and time (1–3: early training, late training, test of trained targets L).

To test the effect of the BoMI modality and its interaction with the target position on performance indices during the reaching of untrained targets, we ran a rANOVA with BoMI modality (1–3: IMUmod, EMGmod, HYBmod) and target position (1–3: L/2, 3L/2, other dir L) as within-subject factors. Since movement time and smoothness depended on the target distance, we did not study the effect of the target position for those metrics.

In order to identify the factors that influenced the planarity of the IMU and the EMG signals, we considered the Variance Accounted For by the first two PCs (i.e. planarity) of

both signals during the test phase and ran two Friedman tests with body-signal type (1–2: IMU signal, EMG signal) and BoMI modality (1–3: IMUmod, EMGmod, HYBmod) as factors. To test the differences in the Classification Accuracy between the control and the auxiliary signal, we ran a Wilcoxon signed rank test in each BoMI modality.

We verified that the assumption of running rANOVA were met by testing the sphericity of the data with the Mauchly's sphericity test and the normality of the data with the Anderson-Darling test. All performance metrics data were normally distributed but were not spherical. Thus, we applied Greenhouse- Geisser correction to the p-values. Post-hoc analyses (Fisher's LSD test for rANOVA and Wilcoxon signed ranked test for Friedman) were carried out to verify statistically significant main effects and interactions. The threshold for significance was set at 0.05. All analyses were performed in Statistica (Statsoft, Tulsa, OK, USA).

3.3 Results

3.3.1 Performance measures

During training, with practice, all participants reached a higher level of control skill. As shown in figure 3.2A, they became faster from first to last training epoch, significantly decreasing the time to reach the targets with each BoMI modality. They also moved the cursor along straighter lines and with increased smoothness in each modality. During the test phase, when reaching targets placed in the trained directions, the participants were able to maintain the performance they acquired during late training with each BoMI modality (also figure 3.2A). Results of the rANOVA model are reported in table 3.1.

Table 3.2 summarizes the results of rANOVA on test targets. In the test phase, the performance while reaching targets placed in untrained positions was significantly different depending on the BoMI modality. This effect was due to the EMGmod being the hardest modality, while performance during IMUmod and HYBmod resulted comparable for every metric. Moreover, the EMGmod performance in terms of movement time and smoothness degraded significantly more than the other modalities when reaching to further targets (figure 3.2B, 3L/2 targets, green bars). The post-hoc analysis is shown in table 3.2.

3.3.2 IMU and EMG contribution in HYB BoMI modality

Figure 3.3A depicts the angle between IMU and EMG components of the cursor position with respect to the peripheral target during the hybrid BoMI modalities, while figure 3.3C shows the norm of the error between the actual cursor position and the EMG and IMU

Table 3.1 Table of effects of rANOVA model on each performance index computed on trained targets.

		F	p_{gg}
MT	Modality	2.13	0.154
	Time	24.44	**<0.001**
	Modality:time	0.08	0.952
LI	Modality	1.93	0.178
	Time	18.40	**<0.001**
	Modality:time	0.37	0.746
MS	Modality	5.15	**0.027**
	Time	23.61	**<0.001**
	Modality:time	0.37	0.715

Table 3.2 Table of effects of rANOVA model on each performance index computed on test targets.

		F	p_{gg}	Fischer LSD	p
MT	Modality	11.3	**0.003**	IMUmod v EMGmod	**0.001**
				HYBmod v EMGmod	**<0.001**
				HYBmod v IMUmod	0.357
	Target	132.4	**<0.001**	L/2 v other dir L, L/2 v 3L/2, 3L/2 v other dir L	**<0.001**
	Modality:target	7.4	**0.003**	IMUmod (3L/2) v EMGmod (3L/2)	**<0.001**
				HYBmod (3L/2) v EMGmod (3L/2)	**<0.001**
				HYBmod (3L/2) v IMUmod (3L/2)	0.420
LI	Modality	4.89	**0.015**	IMUmod v EMGmod	**0.021**
				HYBmod v EMGmod	**0.007**
				HYBmod v IMUmod	0.639
	Target	6.5	**0.005**	L/2 v other dir L	**0.02**
				L/2 v 3L/2	**0.002**
				3L/2 v other dir L	0.299
	Modality:target	0.74	0.572		
MS	Modality	18.7	**<0.001**	IMUmod v EMGmod	**<0.001**
				HYBmod v EMGmod	**<0.001**
				HYBmod v IMUmod	0.265
	Target	99.61	**<0.001**	L/2 v 3L/2, 3L/2 v other dir L	**<0.001**
				L/2 v other dir L	**0.006**
	Modality:target	12.69	**<0.001**	IMUmod (3L/2) v EMGmod (3L/2)	**<0.001**
				HYBmod (3L/2) v EMGmod (3L/2)	**<0.001**
				HYBmod (3L/2) v IMUmod (3L/2)	**0.002**

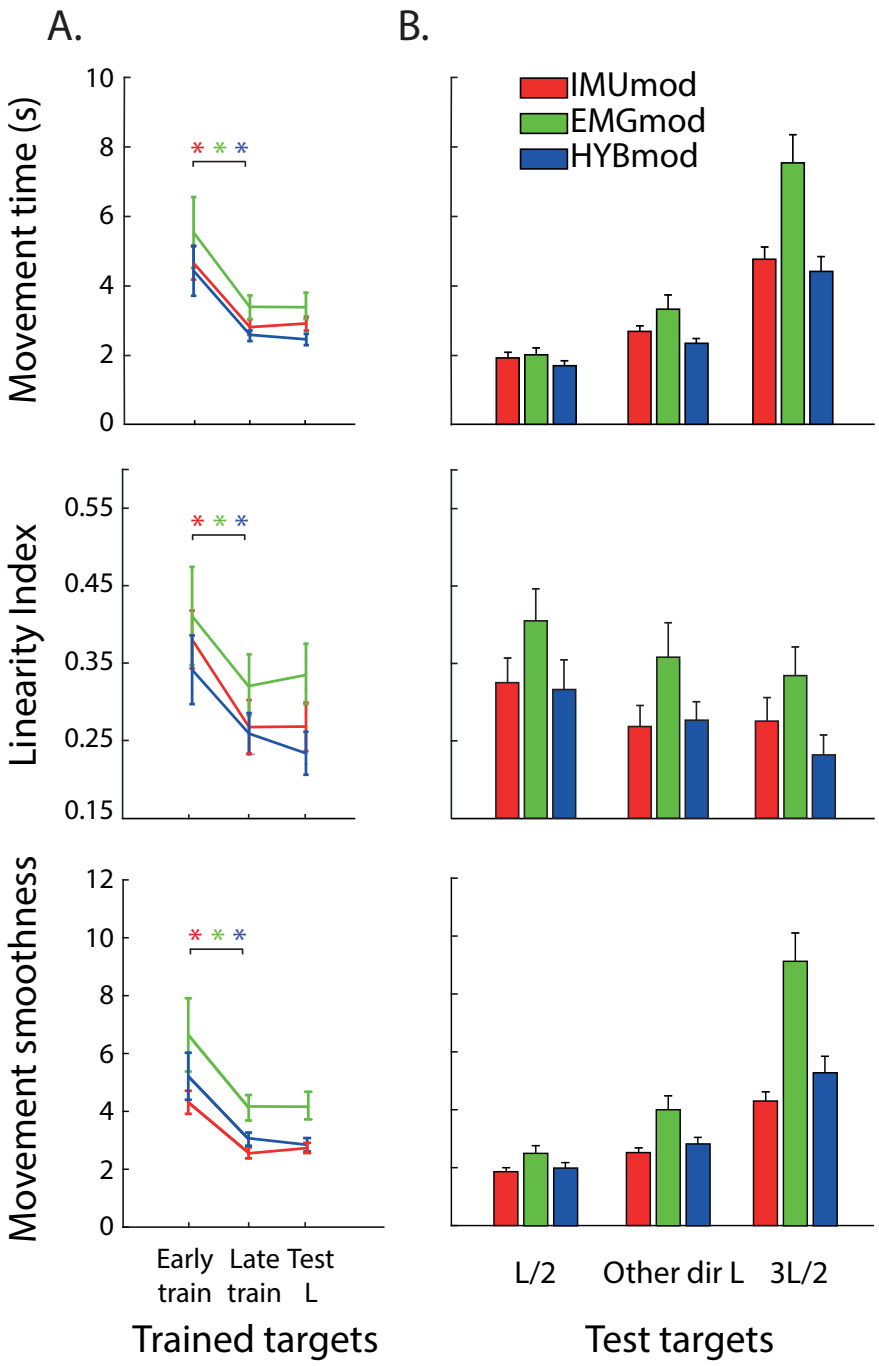

Figure 3.2 Panel A: Performance measures while reaching targets placed in the trained directions (30°, 0°) at distance L from HOME during early, late training and test with the three BoMI modalities. Panel B: Performance measures while reaching targets placed in the untrained positions (trained directions at distances L/2 and 3L/2 and untrained directions 15° at distance L) with the three BoMI modalities. Red lines refer to IMUmod, green lines to EMGmod and blue lines to HYBmod. Mean across participants and targets is plotted with standard error. Asterisks represent significant differences as follows: * $0.01 < p < 0.05$, ** $0.001 < p \leq 0.01$, *** $p \leq 0.001$.

components. While the two components started approximately from the same position (close to the HOME target), they steadily diverged with the maximum reached at the end of the reaching movement (figure 3.3A). Interestingly, about halfway through the movement, the norm of the EMG error component became stationary (figure 3.3C), while the norm of the IMU error decreased monotonically throughout the end of the movement. As expected, the frequency analysis proved that the EMG was the noisiest component (figure 3.3D). An example of cursor trajectory for a single reaching movement of one participant is shown in figure 3.3B.

Furthermore, the velocity analysis showed that the EMG component of the cursor during the hybrid BoMI modality epochs reached higher speed than the HYB and the IMU component (figure 3.4A). Not only the EMG presented the higher speed, but it also gave the higher contribution during each hybrid BoMI epoch (figure 3.4B). This latter condition was consistent with the EMG trajectory being the closest to the actual hybrid cursor throughout the entire movement (figure 3.4C).

3.3.3 Control strategies within BoMI modality

Figure 3.5 summarizes the dimensionality of both IMU and EMG signals during the test phase of each BoMI modality. The variance accounted for by the first two PCs of both signals was consistent across BoMI modalities ($p = 0.627$). However, we found the redundancy of the EMG signal to be inferior to that of the IMU, as a larger amount of variance was explained by its first two components ($p = 0.02$). Post-hoc analysis revealed that this effect was consistent for each modality (IMUmod: $p = 0.018$—figure 3.5A, EMGmod: $p = 0.048$—figure 3.5B, HYBmod: $p = 0.026$— figure 3.5C). This suggests that EMG signals have a stronger planar (*i.e.* linear) structure than IMU signals.

Figure 3.6 summarizes the Classification Accuracy (CA) for the six classifiers trained using EMG and IMU endpoints during the late training phase in each control modality when discriminating among endpoints (or equivalently, among the 3 trained targets) during the corresponding modality in the early training phase. The analysis on body endpoints revealed that the accuracy of a classifier trained on each BoMI modality was the greatest when based on the control signal. Specifically, a classifier trained on IMU endpoints in the late training phase of IMUmod would discriminate targets with significantly higher accuracy than one trained in the same conditions but on EMG endpoints ($p < 0.001$, figure 3.6A). Similarly, a classifier trained on EMG endpoints in the late training phase of EMGmod, would distinguish among targets significantly better than one trained on IMU endpoints ($p < 0.001$, figure

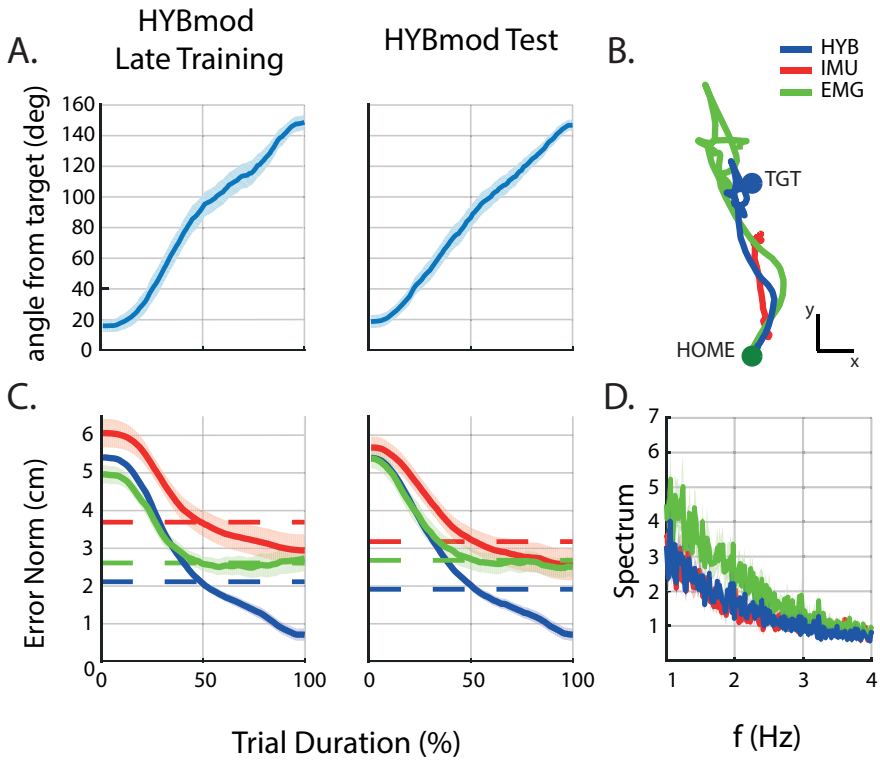

Figure 3.3 Panel A: Angle between the IMU and EMG components of cursor positions with respect to the peripheral target during hybrid BoMI modalities (late training and test). Panel B: Cursor trajectory of a single reaching movement of one participant during test session of HYBmod. Cursor (blue) is shown together with its IMU (red) and EMG (green) components. Panel C: Norm of the error vector between the peripheral target and the HYB (blue), IMU (red) and EMG (green) components of the cursor position respectively. Dashed lines represent error norm values at 50% of trial duration. Panel D: spectrum of cursor (blue) and its IMU (red) and EMG (green) components during test session of HYBmod. Panels A, C and D show mean values across repetitions, targets and participants against the standard error.

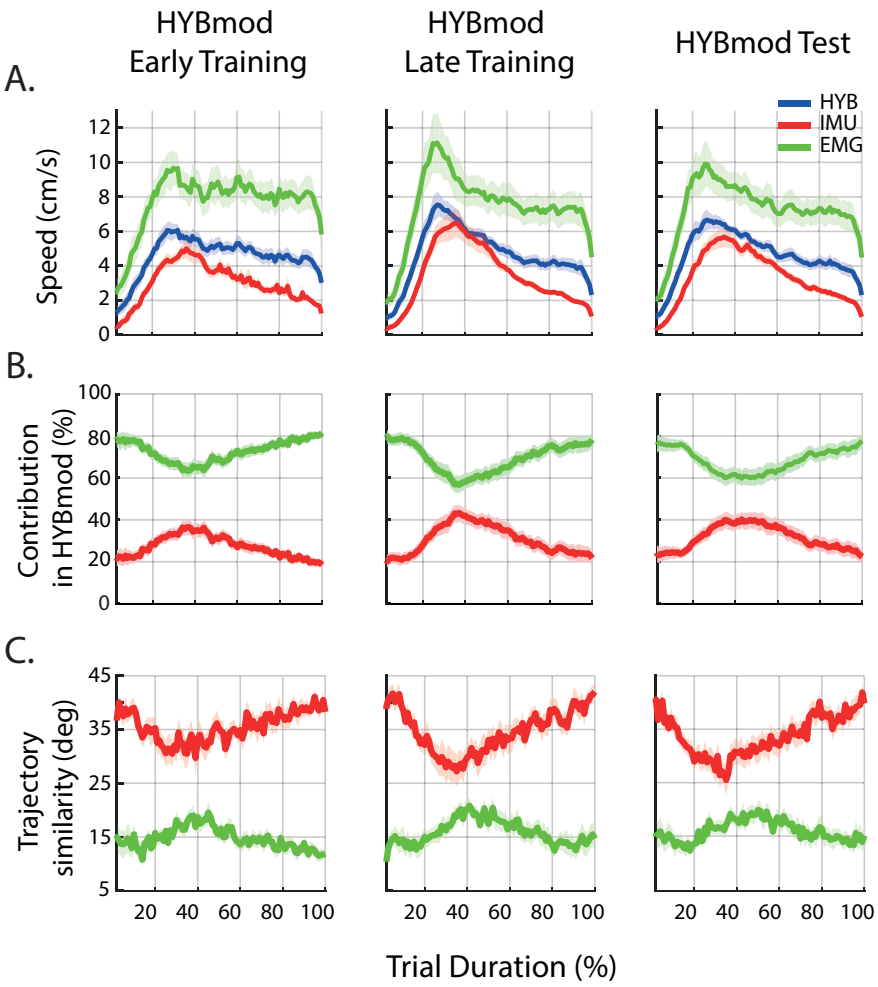

Figure 3.4 Panel A: Speed of the HYB (blue), IMU (red) and EMG (green) component of the cursor. Panel B: Contribution of IMU (red) and EMG (green) component in the velocity of the cursor. Panel C: Angle values between v and v_{IMU} (red) and between v and v_{EMG} (green). All conditions are shown during hybrid BoMI modalities (early, late training and test). Mean across repetitions, targets and participants is plotted against its standard error.

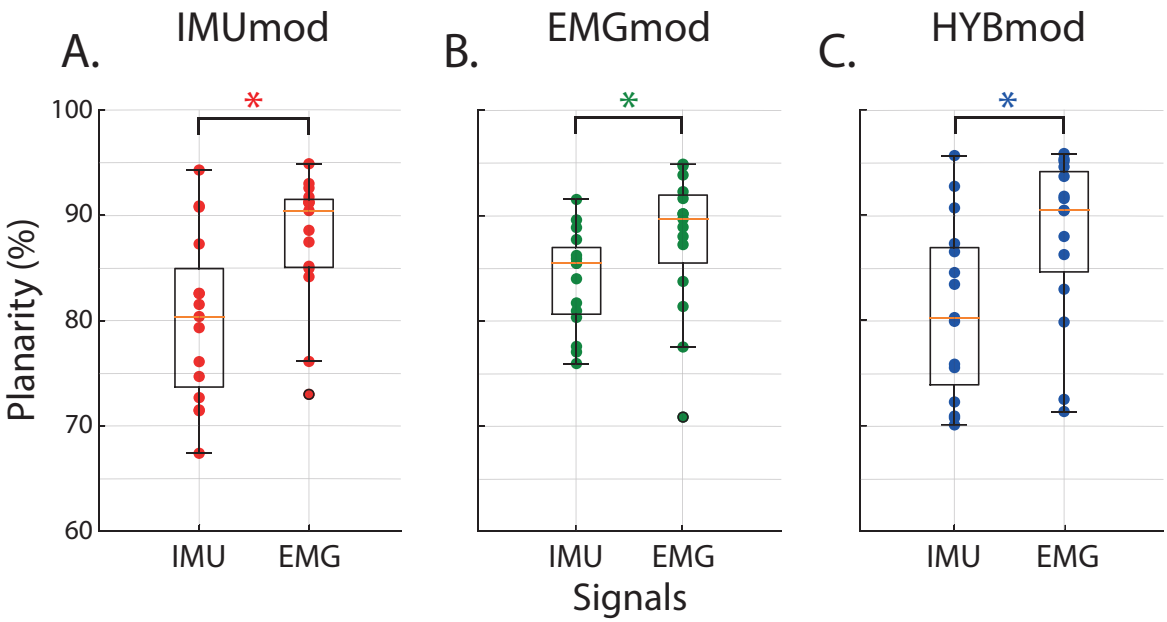

Figure 3.5 Planarity values for IMU and EMG are shown for each BoMI modality (Panel A: IMUmod, red dots; Panel B: EMGmod, green dots; Panel C: HYBmod, blue dots). Boxplots are showing first quartile, median (yellow), third quartile and outliers. Asterisks represent significant differences as follows: * $0.01 < p < 0.05$, ** $0.001 < p \leq 0.01$, *** $p \leq 0.001$.

3.6B). Figure 3.6 shows the accuracy of the two classifiers in HYBmod also. Since in this modality both IMU and EMG signals contributed to cursor control, resulting in no distinction between control and auxiliary signal, we did not run any statistical analysis. It is, however, interesting to note that CA values were in between those obtained during IMUmod and EMGmod. Figure 3.7 depicts the CA of the classifier trained on IMU endpoints during the late training phase of IMUmod and tested on both IMUmod and HYBmod. Since the regression was transforming EMGs into IMU signals, we wanted to evaluate whether IMU endpoints in IMUmod and HYBmod were comparable. As mentioned earlier, IMU endpoints between early and late training phases of IMUmod were consistent (figures 3.6A and 3.7, 1st and 2nd column). Interestingly, kinematic endpoints between IMUmod and HYBmod differed, since classification accuracy values dropped (figure 3.7, 3rd and 4th column).

3.4 Discussion

This study delivered three main findings: (i) the proposed hybrid control led to comparable performance to IMU-based control and significantly outperformed the EMG-only control; (ii) hybrid cursor control was predominantly influenced by EMG signals, indicating that

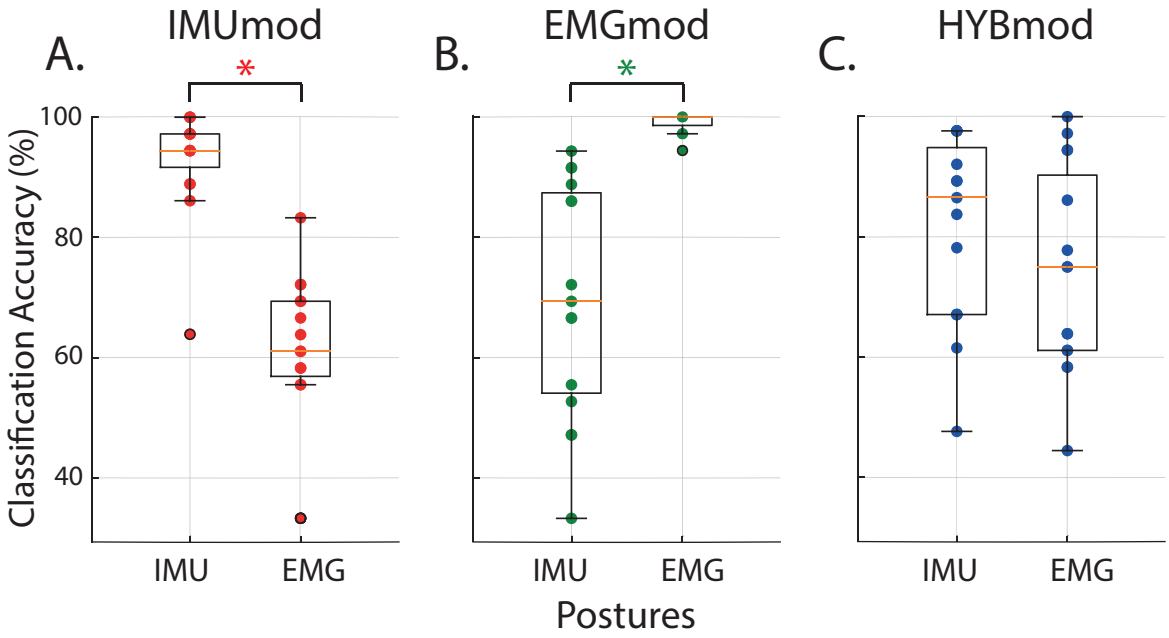

Figure 3.6 IMUs and EMGs endpoints classification accuracy for IMUmod (Panel A, red dots), EMGmod (Panel B, green dots) and HYBmod (Panel C, blue dots). Each dot represents the CA of an individual participant. Boxplots are showing first quartile, median (yellow), third quartile and outliers. The higher the accuracy in a specific modality, the more similar the endpoints of that body signal between early and late training. Asterisks represent significant differences as follows: * $0.01 < p < 0.05$, ** $0.001 < p \leq 0.01$, *** $p \leq 0.001$.

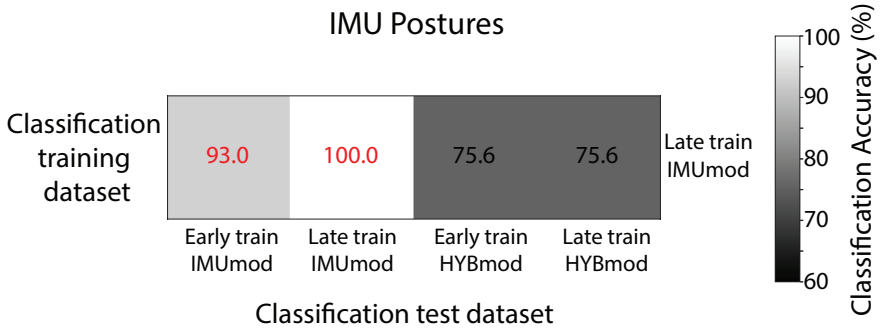

Figure 3.7 Confusion matrix of classification accuracy for the classifier trained with IMU endpoints on the late training phase of IMUmod. The classifier has been tested on four conditions: during early and late training phases of IMUmod (1st and 2nd column) and during early and late training phases of HYBmod (3rd and 4th column). The higher the accuracy, the more similar the kinematic endpoints between the late training phase of IMUmod and the phase shown in each column.

combining EMG with IMU signals allows to efficiently target muscle activations, overcoming the limitation of a EMG-based control; (iii) participants were able to effectively identify the physical signals that actually mattered for the cursor control.

3.4.1 Relevance to rehabilitative therapy

Our novel hybrid control led to the best performance along with the IMU control, confirming our initial hypothesis of motion-preferable control. Thus, our hybrid EMG-IMU control would allow implementing a rehabilitative therapy using joint movements and muscle activities, while at the same time avoiding the performance loss associated with EMG-based control alone. Indeed, in our study, EMG was the least efficient BoMI modality, especially when reaching distal targets. This implies that, while controlling with EMG alone, participants were required to exert and maintain a greater muscle contraction to reach the more distal targets, resulting in a less efficient control. The hybrid modality solved this issue by including the IMU signals in the control scheme, while still maintaining the EMG contribution.

The possibility to directly target muscles for tackling abnormal muscle activations could be achieved by making modifications to the BoMI control map that accounts for EMG contribution (i.e. WEMG). Specifically, we could facilitate or discourage the activation of a targeted muscle by assigning a reward/penalty weight to it. The same approach had already been tested with a motion-based BoMI, and it was shown to successfully increase the strength and range of motion of the body side that was more impaired following SCI (Pierella et al., 2015).

Hybrid control was designed to transform EMG envelopes into control signals equivalent to IMU signals so as to obtain smoother transitions between modalities. Interestingly, despite the fact that IMUmod outperformed EMGmod, the EMG signal contributed the most in the hybrid BoMI modality (figure 3.4B). This is additional evidence to validate the rehabilitative potential of the proposed interface, as it demonstrates the prevalent role of muscles in the hybrid control scheme. The EMG input was associated with a higher gain in the hybrid control as it was quickly driving the cursor towards the target (figure 3.3C). On the other hand, the IMU had lower gain but also lower noise (figure 3.3D). Its main effect was to smoothen the noisier EMG contribution. Thus, hybrid control seems to combine the best of both signals, by exploiting the faster EMG signal in approaching the target and the damping effect of the IMU to reach the target accurately.

We want to note that, while this was a single-day study, use of the interface for assistive as well as rehabilitation purposes will span multiple days. This will inherently pose several

challenges, such as the necessity to reposition the IMU and EMG sensors in a similar fashion during consecutive days. Moreover, the main control maps for IMU and EMG-based control, as well as the polynomial regression model needed for the hybrid control, might require day-to-day recalibration. Several techniques have been developed to mitigate the effect of inconsistent electrode placement and recalibration-need in a multi-day sessions scenario (Farshchian et al., 2018). A future iteration of this study should consider integrating those procedures to improve the stability of the interface during the course of a rehabilitation protocol.

3.4.2 Relevance to basic and transactional neuroscience

While developing new technology to be applied in clinical rehabilitation is a leading motivation for this work, Body-Machine Interfaces have a domain of potential application that includes but is not limited to clinical use. As an example, BoMIs have been used to control a variety of devices, such as drones (Miehlbradt et al., 2018) and robotic arms (Ajoudani et al., 2012; Jain et al., 2015). Furthermore, there is an important application of this instrument to deepen the understanding of the human motor system. While there is an evident correlation between EMG activity and movement, understanding the causal connection between the two is hampered by complex cascade of neuromuscular, sensory, and biomechanical elements that intervenes between the observed electrical activity and the related limb motions. In this respect, the BoMI offers a unique method to artificially enforce EMG causality on the observed and controlled motion without intervening musculoskeletal dynamics. Our findings suggests (e.g. figure 3.3) that in this modified causality context, the users gave a leading role to the direct EMG control in the initial part of the movement, while the second phase of the movement, where the cursor was performing the final approach to the target, was driven by the IMU motion signals, which were affected by musculoskeletal dynamics. This can be taken as evidence that the nervous system exploits mechanical properties of the moving body to facilitate the control in the final reaching component. Therefore, the test of the hybrid BoMI in unimpaired participants served the specific purpose of providing a normative baseline for understanding of how these different control signals are organized by the motor control system. This is an essential step to evaluate performance in a population with SCI or other neurological conditions.

3.4.3 Control strategies within BoMI modality

Numerous studies suggested that our brain, in order to generate actions designed to realize a desired task, develops an internal representation (i.e. the internal model) of the task as the evolution of a dynamical system (Donchin et al., 2003). A recent study from our group (Pierella et al., 2019) mathematically described how, in the BoMI context, users not only generate an inverse model of the control map, but that their inverse model converges as a first-order exponential toward a particular state (i.e. a single solution). In this study, we noticed a similar condition. Specifically, participants were required to learn three different inverse models to effectively control the cursor with each of the three different BoMI modalities. When controlling with IMU or EMG signals alone, they succeeded at identifying the physical signals that actually mattered for cursor control, and to distinguish the primary control signal from the auxiliary signal. In other words, the kinematic and muscular endpoints that they held were consistent throughout the training with IMUmod and EMGmod respectively. Thus, in each modality participants converged to an inverse solution of the BoMI mapping EMG or IMU signals onto cursor position. Conversely, the analysis we used could not prove whether participants converged to an inverse solution with HYBmod as well, since in this case there was no distinction between the control and the auxiliary signal. However, it is interesting to note that in HYBmod, with the EMG signals being mapped into the IMU space, participants changed their kinematic endpoints with respect to the IMUmod (figure 3.7). This is not trivial, as participants could have merely replicated the kinematic endpoints they held in IMUmod. Hence this suggests that muscle activity was greatly contributing to the hybrid control, further supporting the applicability of the interface for rehabilitative purposes.

Our initial hypothesis was that kinematic and muscular endpoints would have consistent behavior due to the physiological correlation between IMUs and EMGs. As such, changes in endpoint in one space should have been reflected in changes in the other, regardless of which was associated with the control signal, and which with the auxiliary signal. However, this discrepancy between the two spaces was indeed present and allowed participants to learn how to operate the interface. As a matter of fact, when participants did not have to rely on the EMG signals as they were operating the interface with the IMU modality, they activated muscles differently while maintaining a similar IMU endpoint at the end of a reaching movement (figure 3.6A). By contrast, when they were involved with the EMG-based control, they showed different IMU endpoints while maintaining consistent muscle activations at the end of a reaching movement (figure 3.6B). We speculate that there is an analogy between results in both conditions and the uncontrolled manifold (UCM) hypothesis (Scholz and Schöner, 1999) and optimal feedback control (Todorov and Jordan, 2002), which state that

the controller (i.e. the brain) allows elements to show higher variability as long as this does not affect the desired value of the task, i.e. as long as the elements are auxiliary signals. Specifically, the concept of redundancy plays a key role in explaining our results. On one hand, it is known that a given task can be accomplished with different movement patterns (Bernstein, 1966) and that a specific movement can be generated by an infinite number of muscle activation combinations (Sharif Razavian et al., 2019). According to the natural redundancy of muscles compared to the number of joints, IMU signals are less redundant than EMGs, thus explaining why participants were able to have different EMG activations while maintaining a similar movement strategy. On the other hand, in our task, the EMG signals had a stronger planar structure than the IMUs. This task-related redundancy may explain why the participants were able to have different IMU endpoints while maintaining similar muscle activations. The co-existence of these two distinct types of redundancy was merely a result of our experimental setup. In fact, we were recording signals from a subset of muscles of the upper body, thus ignoring others that could have significantly contributed to the overall EMG dimensionality.

3.4.4 Limitations

We also need to highlight some possible limitations of our hybrid approach. In order to get a smooth transition between IMU and EMG control modalities, a stable regression between those signals is required. Here, we addressed the problem of predicting IMUs from EMGs with a non-linear regression model to accommodate for the natural nonlinear relationship existing between movement and muscle signals. A general approach to handle nonlinear problems is to add more polynomial features (e.g. two in our study) as predictors. Adding polynomial features is simple to implement and can be effective, but the polynomial order to use in the model might be difficult to validate. In this sense, a non-parametric kernel-based regression (Härdle and Vieu, 1992) might be more appropriate to depict this relationship without having to explicitly specify the polynomial order. Furthermore, our approach did not consider the temporal dependency between EMG and IMU. Since EMG is anticipating IMU from a temporal point of view, a model that considers this timedifference might be more robust (e.g. a recurrent neural network (Lipton et al., 2015)). Despite these limitations, we found our particular instance of regression to work properly with unimpaired participants, thus constituting a promising starting point with the potential to be further fine-tuned to increase its robustness.

In previous sections, we discussed how performance with EMG modality resulted in inefficient control especially when reaching the more distal targets, due to the requirement of higher muscle contractions. Note that the control of the cursor was position-based. The exertion of those uncomfortable contractions could have been avoided with a velocity-based control. This could have improved performance during the EMG-only control (Segil et al., 2015), but it could have, nonetheless, harmed performance when reaching other portions of the workspace. In this sense, more extensive tests should be carried out, possibly combining our hybrid approach with a velocity-based cursor.

Finally, we want to make one cautionary note regarding our computation of the experimental redundancy. It was defined as the variance accounted for by the first two principal components derived from PCA. While for each BoMI modality the forward map was linear, the control strategy might have been linear or non-linear. While a linear strategy corresponded to a planar manifold embedded within the body signals, a nonlinear strategy corresponded to a curved manifold. Since PCA is a linear algorithm approximating a participant's control strategy with a planar manifold, a nonlinear control strategy would result in signals whose variance cannot be fully accounted for by linear combinations of two principal components. This led to an approximated computation of the experimental redundancy (or planarity, equivalently), and it is a known particular instance of the more general observation that PCA, when applied to a nonlinear dataset, tends to overestimate the dimensionality of the data (Tenenbaum et al., 2000).

3.5 Conclusions

The proposed interface sets the ground for an unsupervised algorithm that concurrently includes motions and muscles in the control-scheme. Our hybrid approach combines the best features of muscular and kinematic signals, by exploiting the former to quickly move the cursor towards the target and exploiting the damping effect of the latter to reach the target accurately. We also found that muscles were predominantly contributing for the cursor control. This indicates how the interface could be used in rehabilitative protocols to efficiently target muscle activations, without the limitations associated with EMG-based control alone. Moreover, our work showed how participants were able to effectively identify the physical signals that actually mattered for the cursor control, suggesting that they converged to an inverse solution of the BoMI mapping EMG or IMU signals onto cursor position.

Chapter 4

Building an adaptive interface via unsupervised tracking of latent manifolds

4.1 Introduction

Understanding how we learn to interact with a machine is of primary importance when designing human-machine interfaces. The ability of the nervous system to reorganize its structure and create novel neural pathways in response to learning is widely recognized (Dayan and Cohen, 2011; Donati et al., 2016; Kandel et al., 2000). Motor training and skill learning rely on neural plasticity in the nervous system enabling each individual to adapt to new environments (Shadmehr and Mussa-Ivaldi, 1994; Shadmehr et al., 2010; Wei et al., 2005), to learn new tasks (Mawase et al., 2017) and to efficiently operate a variety of devices (Danziger et al., 2009). On the other hand, to optimize the human interactions with devices, it is necessary to design interfaces that can understand the user's abilities, preferences, and intentions.

The approach followed by brain machine interfaces (BMIs) (Carmena et al., 2003; Shenoy and Carmena, 2014) is to decode intended actions from high dimensional neural recordings, and then convert (or encode) the decoded intentions into lower dimensional set of commands for operating a device (*e.g.*, moving a computer cursor or robotic manipulator). Similarly,

The content of this chapter has been published as Rizzoglio, F., Casadio, M., De Santis, D. and Mussa-Ivaldi, F.A. 2021. Building an adaptive interface via unsupervised tracking of latent manifolds. *Neural Networks*.

body machine interfaces (BoMIs) extract a low dimensional control space from more downstream information related to the execution of voluntary movements, as body kinematics and muscle activity (Casadio et al., 2012; Farshchiansadegh et al., 2014; Miehlbradt et al., 2018; Rizzoglio et al., 2020).

The efficiency of an interface in transferring the user's input to the device is tightly dependent on the joint user-decoder performance. There are at least two main ingredients that make the acquisition of interface control proficiency possible, i) user learning and ii) decoder adaptation.

In a typical BMI (Shanechi, 2016) the decoder is initialized by a supervised calibration procedure, which creates a map based on labelled examples of desired actions and corresponding neural activities. However, a good calibration is not sufficient to achieve proficient BMI control, as the performance of the decoder often degrades and fluctuates when evaluated online, due to imperfect predictions, recording instabilities, changes in neuronal properties, attentional changes, and changes brought about by learning (Barrese et al., 2013; Downey et al., 2018). BoMI decoders suffer from similar issues. The decoder of a BoMI, also referred to as "forward map", is initialized by unsupervised identification of the low-dimensional, latent manifold of unconstrained users' movements recorded during an initial calibration (Casadio et al., 2010). This latent manifold is expected to change with subsequent practice. Therefore, a discrepancy is likely to develop in time between the evolving latent manifold of the user and the initial BoMI forward map. Indeed, it has been observed that extensive practice with a BoMI led to the consolidation of task-specific movement strategies (Pierella et al., 2017b) and several studies in brain machine interfaces demonstrated the existence of a stable manifold of neural activity linked to BMI use and interpreted this as the result of neural adaptation following extensive practice with the interface (Gallego et al., 2020; Ganguly and Carmena, 2009; Oweiss and Badreldin, 2015; Shenoy and Carmena, 2014).

The consequence of adopting a fixed activity-intention map after its initial offline tuning, is that the user is left with the burden of learning how to use the interface for achieving new goals within new operating conditions that the decoder has not been optimized on.

Closed-loop supervised decoder adaptation in BMIs has been proposed as an effective way to increase decoder performance during use. This was the case when allowing the decoder parameters to smoothly change according to the inferred movement goal during within-session interface operation (Dangi et al., 2013; Orsborn et al., 2014). A very recent study considered instead the case of performance loss arising from decoder instabilities across sessions. They showed that by calibrating the decoder using features within the manifold of stable activity of the recorded neurons, performance of the BMI can be reliably recovered by

manifold alignment across sessions (Degenhart et al., 2020). Despite contributing encouraging results, these current perspectives rely either on the knowledge of user intent or on the presence of an established manifold of neural activity linked to the use of the BMI. Hence, they cannot be easily extended to facilitate the use of the interface when the movement goal is unknown or when the formation of new neural strategies is still ongoing and the activity manifold has not yet consolidated (Oby et al., 2019).

Given the current limitations of closed-loop decoder adaptation, we propose a procedure for facilitating interface operation that does not rely on estimates of user intent and can be applied from the very initial stages of learning. The procedure is initially developed for application with body machine interfaces and exploits a non-linear autoencoder (AE) network (Kramer, 1991) trained iteratively to identify and track the evolution of the latent manifold of its inputs.

Previous work from our group suggested that individuals training with a linear BoMI that was adapted iteratively based on movement statistics increased movement efficiency compared to a fixed interface and were able to develop a more faithful internal representation of the BoMI forward map (De Santis et al., 2018; De Santis and Mussa-Ivaldi, 2020). Building on these results, the hypothesis of this study is that the operation of the interface will be facilitated if the BoMI forward map is updated online so as to match the evolving manifold of the user's input. That is, in other words, if the output manifold of the forward map and the user's movement manifold were isomorphic.

The major advantages of the proposed approach are that it can be applied in the earliest stages of interface operation, it does not require interrupting the operation of the interface, and it does not need to incorporate information about the state of the task, in contrast with other adaptive interfaces described in the literature (DiGiovanna et al., 2008; Mahmoudi et al., 2008; Sanchez et al., 2009). Moreover, the use of a non-linear algorithm allows representing a curved manifold that is more consistent with the variance structure of input signals (Portnova-Fahreeva et al., 2020) compared to a linear approximation.

We decided to test our hypothesis within the framework of body machine interfaces, as they allow evaluating learning of a low-dimensional manifold in a more immediate way, *i.e.* without requiring spike sorting or other procedures necessary for preprocessing neural activity. However, there has been only preliminary evidence regarding the applicabilty of non-linear AEs to the control of low dimensional devices (Pierella et al., 2018; Vujaklija et al., 2018). Hence, the results of this work will also serve as validation for the use of AEs in BoMI applications.

The following sections are structured as follows.

We first describe some preliminary concepts (§4.2 Preliminaries). We introduce the framework of the body machine interface (§4.2.1) and how it can be implemented using non-linear AE (§4.2.2). At this stage, we assume the interface to be fixed after calibration (*i.e.*, fixed autoencoder).

Then, we propose an implementation of iterative learning of a non-stationary latent manifold and we call this adaptive autoencoder (a-AE) (§4.3.1). Finally, we give details of the two experiments we carried out for validating the performance of the a-AE both offline (§4.3.3) and online (§4.3.4). In the first study, we performed a sensitivity tuning to identify the hyperparameters that allow the a-AE to smoothly track a non-stationary movement distribution. This operation was conducted offline on real data collected from a naive BoMI user performing a reaching task with a fixed BoMI. The second study was conducted with the purpose of testing if training with an adaptive autoencoder facilitates user adaptation to the interface when compared to a fixed BoMI. In other words, we verified whether human-machine interaction might benefit if modelled as a bi-directional adaptation (*i.e.*, co-adaptation) of a learning brain with a learning interface. During this online test, the a-AE hyperparameters were set as those that yielded the best performance in the first study.

4.2 Preliminaries

4.2.1 Body machine interfaces

BoMIs (Casadio et al., 2012) are a class of human-machine interfaces that provide a link between human body motions and an external machine. The general purpose of a BoMI is to enable its user to retain a complete or shared control over the device through signals derived from the user's body. The BoMI exploits the assumption that the signals needed to control the external device lie on a low-dimensional manifold embedded within the higher-dimensional body signal space. In a BoMI, sensors (*e.g.*, infrared markers (Casadio et al., 2010), inertial measurement units (Farshchiansadegh et al., 2014), electromyographic electrodes (Rizzoglio et al., 2020)) are placed on the upper body of the user to record a d-dimensional vector of body signal, q. A forward map $f(.)$ is then constructed to transform q into an m-dimensional control vector, p, (with $m < d$) encoding the commands for controlling an external device (*e.g.*, computer cursor, powered wheelchair, quadcopter):

$$p = f(q) \tag{4.1}$$

The BoMI forward map f is determined in a calibration session, where BoMI users are asked to freely move their limbs so as to explore their range of motion. This procedure is not task-related, as users do not receive visual feedback of their movements and are not yet connected to the device. Data collected during calibration are then used as training set for a dimensionality reduction (DR) algorithm that derives f by extracting the low-dimensional, m-D, manifold in which the highest amount of variance of body signals is available. Common iterations of BoMIs exploit linear DR methods to extract such low-dimensional manifold, *e.g.* Principal Component Analysis (PCA) (Wold et al., 1987), or Kalman Filter (Seáñez-González et al., 2016). In this study, we explore the use of a non-linear DR algorithm, autoencoder networks.

4.2.2 Fixed autoencoder

AEs are unsupervised artificial neural networks capable of learning efficient lower dimensional representations of the input data without labelled data. An AE is a cascade of two components: an encoder $\boldsymbol{E}$ that converts the inputs to a lower dimensional latent representation, or code, followed by a decoder $\boldsymbol{D}$ that converts the latent representation into the outputs, with the same dimensions as the inputs. An AE attempts at learning to perform the identity map, copying its input, $\boldsymbol{q}$, to its output, $\hat{\boldsymbol{q}}(\boldsymbol{w},\boldsymbol{b})$, with minimum loss, thus learning an efficient internal representation of the inputs within their latent manifold. This is achieved by minimizing the mean squared error between $\boldsymbol{q}$ and $\hat{\boldsymbol{q}}$

$$\boldsymbol{J}(\boldsymbol{q},\hat{\boldsymbol{q}}) = \frac{1}{k}\sum_{i=1}^{k}\sqrt{(\hat{\boldsymbol{q}}_{\boldsymbol{i}} - \boldsymbol{q}_{\boldsymbol{i}})^2} \tag{4.2}$$

with k being the number of samples of $\boldsymbol{q}$. $\boldsymbol{w}$ and $\boldsymbol{b}$ are the parameters of the AE, representing the matrices of weights and biases of each layer of the network, respectively. The update rule for the AE parameters is:

$$\boldsymbol{w} \leftarrow \boldsymbol{w} - \alpha\nabla_w\boldsymbol{J}, \qquad \boldsymbol{b} \leftarrow \boldsymbol{b} - \alpha\nabla_b\boldsymbol{J} \tag{4.3}$$

In this study, we used an AE with five layers (see Fig. 4.1) and Batch Gradient Descent (BGD) (Ruder, 2016) to minimize $\boldsymbol{J}$. As for the AE proposed by Kramer (Kramer, 1991), we chose a nonlinear activation function for the hidden layers and a linear activation function for the code and output layers. We did not opt for a deeper network because we aimed at employing a parsimonious architecture. An AE fed with the n-dimensional vector containing the body-signals ($\boldsymbol{q}$) applies the following transformations:

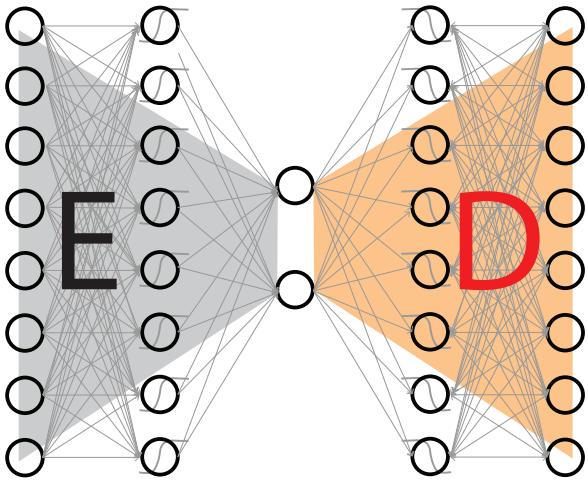

Figure 4.1 Autoencoder (AE) network structure used in this study. The encoder part was a densely connected network that transformed an eight-dimensional input layer into a first hidden layer of the same dimensionality and subsequently a two-dimensional code layer. The decoder transformed the 2D latent space into a second 8D hidden layer and finally back to an 8D output layer. The first and second hidden layers applied a nonlinear transformation (*i.e.*, hyperbolic tangent) to their inputs

$$\boldsymbol{p} = \boldsymbol{E}(\boldsymbol{q}) = \boldsymbol{w_2}(tanh(\boldsymbol{w_1 q} + \boldsymbol{b_1})) + \boldsymbol{b_2} \tag{4.4}$$

$$\hat{\boldsymbol{q}} = \boldsymbol{D}(\boldsymbol{p}) = \boldsymbol{w_4}(tanh(\boldsymbol{w_3 p} + \boldsymbol{b_3})) + \boldsymbol{b_4} \tag{4.5}$$

where $\boldsymbol{w_i}$ and $\boldsymbol{b_i}$, $i = 1:4$ are the matrices of weights and biases of the i^{th} layer, while $\hat{\boldsymbol{q}}$ is the n-dimensional vector containing the body-signals reconstructed by the AE.

In the BoMI context, the encoder sub-network $\boldsymbol{E}$ can be used as a forward map in Eq. 4.1. Thus, the vector $\boldsymbol{p}$ includes the commands for the control of the m-dimensional external device. Importantly, unlike other nonlinear methods for DR (*e.g.*, Isomap (Tenenbaum et al., 2000)), after training the AE on a data set, the encoder $\boldsymbol{E}$ projects new incoming data on the same latent manifold derived from the training data.

4.3 Methods

4.3.1 Adaptive autoencoder

Here we describe the algorithm to adapt the interface accordingly to the evolving manifold of the user's input. We refer to this algorithm as adaptive autoencoder (a-AE).

The a-AE consists of a memory element and a learning element (Fig. 4.2). The memory element stores samples of the data depicting the state of the user (*e.g.*, body motion). The learning element intervenes with a certain frequency to update the parameters of the AE by maximizing its fitness to the data stored in the memory. The parameters that govern the response of the a-AE are:

- f_u, the frequency of learning element updates. An update iteration n is performed every $1/f_u$ seconds.
- k, the size of the memory element, or equivalently the length of the training batch available at each update iteration n - $\boldsymbol{q}^{(n)}$, the *history batch*.
- β, the number of BGD steps over the history batch $\boldsymbol{q}^{(n)}$. Increasing β increases the goodness of fit of the a-AE on the history batch.
- α, the learning rate parameter, which determines the size of each step taken in the direction of the gradient of J (Eq. 4.3). Higher values of α accelerate convergence towards the minimum of J.

Every $1/f_u$ seconds, the learning element triggers an update iteration n to adapt the AE parameters to the history batch stored in the memory. Namely, it runs β BGD steps over $\boldsymbol{q}^{(n)}$ as in Eq. 4.3. To facilitate the a-AE tracking of the user's latent manifold, the AE parameters are updated via transfer learning (Bengio, 2012). Namely, at the start of each update iteration, the parameters of the AE are set to be equal to those of the previous iteration. In the generic n update iteration, the history batch $\boldsymbol{q}^{(n)}$ consists of k samples. If $k \geq f_s/f_u$, where f_s is the sampling frequency of q, then $k-(f_s/f_u)$ are samples from the preceding update step and (f_s/f_u) are new samples. Instead, if $k < f_s/f_u$ there will be no overlap between successive training batches. The pseudocode of the a-AE is shown Table 6.1. The block diagram of the a-AE that we tested online is shown in Fig. 4.2.

4.3.2 Experimental methods

This study consisted of two experiments. The first was an offline test of the proposed adaptive algorithm. The results of this experiment were used to evaluate the adaptive algorithm tracking performance and its sensitivity over the choice of the learning parameters. In the second experiment, we tested the adaptive algorithm by closing the loop of the BoMI. Specifically, we compared static and co-adaptive non-linear BoMIs when controlling the two coordinates of a computer cursor using body signals derived from motions of the upper

Table 4.1 Pseudo-code for the a-AE

Input: k, α, β, f_u, $n = 1$, $t_{elapsed} = 0$
 # Initialize $AE^{(n)} = \{E^{(0)}, D^{(0)}\}$ *with calibration data set* $\boldsymbol{q}^{(0)}$
$AE^{(0)} = f(\boldsymbol{w}^{(0)}, \boldsymbol{b}^{(0)})$, $\boldsymbol{q}^{(1)} = \boldsymbol{q}^{(0)}$
while not terminated **do**:
  # 1. Memory element: update the history batch $\boldsymbol{q}^{(n)}$:
    $\boldsymbol{q}^{(n)}$ = append($\boldsymbol{q}^{(n)}$, $q^{(i)}$)
    if size($\boldsymbol{q}^{(n)} > k$), pop($\boldsymbol{q}^{(n)}[0]$)
  # 2. activate the Learning element
    if $t_{elapsed} = 1/f_u$
      # Step I. Transfer learning on $\boldsymbol{w}$ and $\boldsymbol{b}$
        $\boldsymbol{w}^{(n)} \leftarrow \boldsymbol{w}^{(n-1)}, \boldsymbol{b}^{(n)} \leftarrow \boldsymbol{b}^{(n-1)}$
        $AE^{(n)} = f(\boldsymbol{w}^{(n)}, \boldsymbol{b}^{(n)})$
      # Step II. Run BGD on $\boldsymbol{w}$ and $\boldsymbol{b}$
        for step in β:
          $\hat{\boldsymbol{q}}^{(n)} = D^{(n)}(E^{(n)}(\boldsymbol{q}^{(n)}))$
          $\boldsymbol{w}^{(n)} \leftarrow \boldsymbol{w}^{(n)} - \alpha \nabla_{\boldsymbol{w}} \boldsymbol{J}(\boldsymbol{q}^{(n)}, \hat{\boldsymbol{q}}^{(n)})$
          $\boldsymbol{b}^{(n)} \leftarrow \boldsymbol{b}^{(n)} - \alpha \nabla_{\boldsymbol{b}} \boldsymbol{J}(\boldsymbol{q}^{(n)}, \hat{\boldsymbol{q}}^{(n)})$
          $AE^{(n)} = f(\boldsymbol{w}^{(n)}, \boldsymbol{b}^{(n)})$
        end
      $\boldsymbol{q}^{(n+1)} = \boldsymbol{q}^{(n)}$
      $n = n + 1$
      $t_{elapsed} = 0$
    end
end
Output: the AE model at current iteration n, $AE^{(n)}$

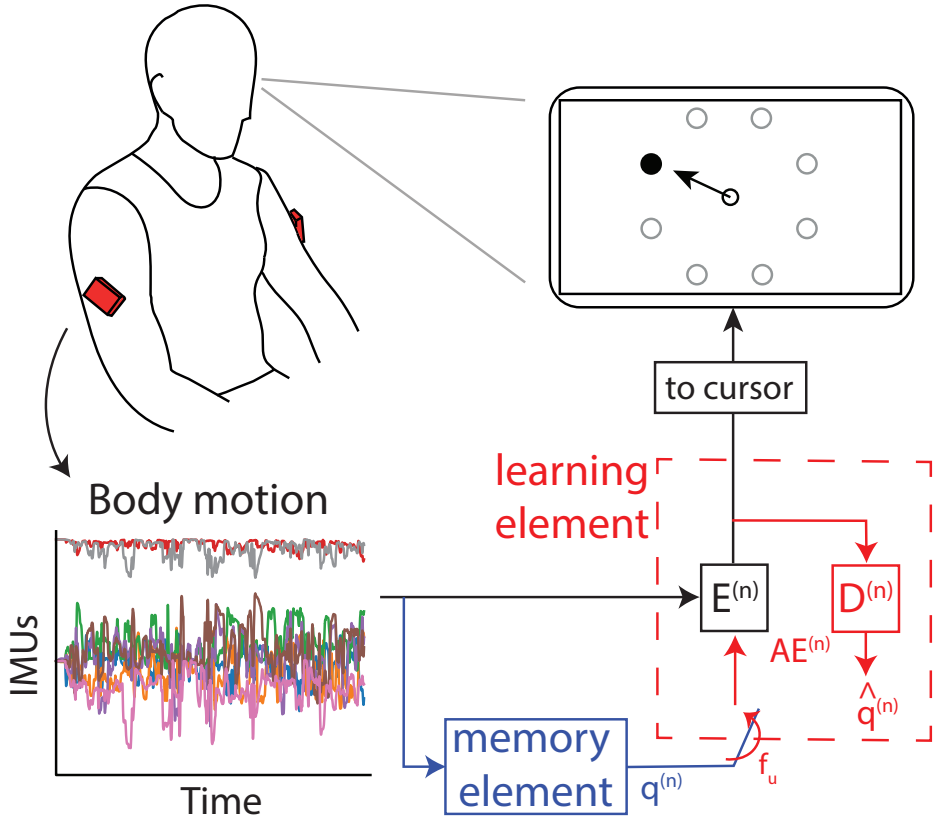

Figure 4.2 Block diagram of the adaptive autoencoder. Body motion recoded by IMUs, shown as orange boxes, is fed to the encoder $\boldsymbol{E}^{(n)}$ and the memory element (blue). $\boldsymbol{E}^{(n)}$ extracts the user's latent space, which is in turn directly converted into coordinates of the computer cursor. Every $1/f_u$ seconds the learning element (red) activates, and the history batch $\boldsymbol{q}^{(n)}$ stored in the memory element is fed through $\boldsymbol{E}^{(n)}$ and $\boldsymbol{D}^{(n)}$ to obtain $\hat{\boldsymbol{q}}^{(n)}$. The parameters of $AE^{(n)}$ are finally updated via minimization of the mean squared error between $\hat{\boldsymbol{q}}^{(n)}$ and $\boldsymbol{q}^{(n)}$.

limbs with an embedding dimensionality of six. We recorded the motion of the arms with two inertial measurement units (IMUs) (3 Space Sensors, Yost Labs, Portsmouth, OH, USA). The sensors were placed bilaterally as shown in Fig 4.2. The IMU sensors used in this study derive orientation in the quaternion format using a complementary filter that integrates raw measurements from accelerometer, gyroscope, and magnetometer. However, we excluded the magnetometer from the IMU complementary filter due to the presence of ferromagnetic structures and transient disturbances that made its measurements unreliable and noisy. We therefore recorded an eight-dimensional movement signal. The BoMI software was custom coded in C#.

4.3.3 Study I: Offline sensitivity tuning

The sensitivity tuning of the a-AE was carried out as an offline simulation using the movement data recorded from a naive individual performing reaching movements with a BoMI. The procedure aimed at selecting the hyperparameter values that would allow the adaptive AE to attain three conditions: (1) providing a stable representation of the user's movement statistics, *i.e.* by converging towards a minimum J over time in Eq. 4.2; (2) minimizing the discontinuity of the interface at the moment of the encoder update; (3) changing at a rate that is neither too slow nor too fast. The discontinuity of the interface is a critical point because it might affect its usability online when operating an external device. For example, in this study, a discontinuous adaptive interface would produce a sudden jump of the computer cursor, which could be perceived as a perturbation and possibly disrupt the user's experience with the BoMI.

Previous studies suggested that the learning curves of users operating a BoMI during a reaching task follow an exponential trend with time constants ranging from approximately 2 to 10 minutes (De Santis and Mussa-Ivaldi, 2020). This means that users display a 50% improvement in control performance only after the first 2 minutes of practice. Hence, it is desirable for the interface to reach a set point within approximately 2 minutes when updated online. Therefore, we set this to be the target value for the network to converge (<1% residual error) during the offline sensitivity tuning. Note that this procedure involved the concurrent tuning of two hyperparameters, the size k of the memory element, and the network learning rate α. In order to estimate which values yielded the most ideal behaviour, we fit a double exponential function to the learning curve derived from each combination of hyperparameters. We used the rate of convergence, defined as the time constant (tau) of the fastest exponential as metric to select the best candidates for the online test according to the rule $5 * tau \sim 120s$.

To find a batch size that was representative of BoMI user's movement dynamic, we tested three values of k referring to short-term (2s), mid-term (20s) and long-term (60s) memory. Previous studies with movement-based body-machine interfaces (Abdollahi et al., 2017; De Santis et al., 2018; De Santis and Mussa-Ivaldi, 2020; Pierella et al., 2017b) have found that naïve users progress from several tens of seconds to complete a reaching movement towards a target to only a few seconds after some practice. Hence, in the context of a short bout ($\sim$15-20 minutes) of reaching task practice with the interface, the three levels of memory provide each update step with the information regarding within trial variation (2s), data distribution when reaching towards a few (1-5) different targets (20s), and information regarding movements distributed over a greater region of space (60s). Then, we chose to test values of learning rate with three distinct orders of magnitude to evaluate which time scale would have allowed the adaptive AE to satisfy the 120s criterion.

In this study, we set β equal to 10 steps of BGD and the a-AE update rate ($1/f_u$) to 2 seconds. The choice of updating the AE parameters every 2s was made to ensure a continuous (and quick) update, that could keep track of the evolving movement manifold of the BoMI user online. Ideally, the update should have been performed as fast as possible. However, we had to consider the time that the preprocessing operations (*e.g.*, filling the memory buffer with the latest kinematic samples recorded) and the training of the AE would have required. Since all these operations took, on average, little less than 1s to be completed, we decided to put some cushion and fix the update rate to 2s, in order to maintain a constant update rate throughout the experiment.

We included the Adaptive Moment Estimation (Kingma and Ba, 2014) optimization to allow meeting this time requirements for online computation of the AE parameters. The a-AE ran in a custom-coded Python thread and all the optimization policies were implemented using Tensorflow (Abadi et al., 2016).

4.3.4 Study II: Online test of the adaptive autoencoder

In the second study we tested the adaptive autoencoder by closing the loop of the BoMI. During this online test, the a-AE hyperparameters were set as those that yielded the best performance in the first study. In our BoMI, the encoder $\boldsymbol{E}$ mapped the eight-dimensional IMU-vector ($\boldsymbol{q}$) into the x-y cursor vector $\boldsymbol{p}$:

$$\boldsymbol{p} = \boldsymbol{E}(\boldsymbol{q}) + \boldsymbol{p_0} \tag{4.6}$$

The offset vector $\boldsymbol{p_0}$ was chosen to make a desired neutral position (*i.e.*, rest position) of the body-space match a corresponding reference position of the cursor. Moreover, the resulting workspace was then rotated and stretched in order to ensure complete coverage of the entire workspace of the cursor (Casadio et al., 2010). While the overall philosophy of the BoMI is to base its operation on a customized mapping, adapted to each individual user, here we had to sacrifice some of this philosophy to the need of obtaining data that could be compared across subjects. This required having all participants start from a common initial condition, *i.e.* from the same initial mapping. To attain a compromise between uniformity and customization, we decided to base the initial BoMI mapping on movement data from a single "representative" user. A single individual (age 25, male), who did not participate in the rest of the study, was asked to freely move the arms exploring the full range of motion for 60 seconds. Movement data, recorded during this calibration phase, were used as a training data set. The encoder derived from this training data set was taken as the initial encoder ($\boldsymbol{E}^{(0)}$). This same initial map was set for all participants.

Participants

We enrolled twenty unimpaired subjects. They did not have any known history of neuromotor or musculoskeletal disorders and exhibited typical joint range of motion and muscle strength. All participants gave their signed informed consent prior to the test. All procedures were carried out in accordance with the ethical standards of the Declaration of Helsinki and Northwestern University IRB approved all human involvement in the study (IRB protocol #STU00057856). Participants were divided in two groups and assigned to one of the following study conditions:

- Fixed map (group **F**, N = 10, age 26.7±6.9, four females): participants were assigned a constant encoder $\boldsymbol{E}^{(0)}$, defined by the common calibration procedure. Therefore, each individual was presented with the same map which did not change throughout the session.

- Adaptive map (group **A**, N = 10, age 26.3±5.9, six females): in this group, the encoder adapted to participants' movements following the a-AE algorithm validated in the first experiment. Initially, the encoder was defined by the same common calibration procedure as in the previous group $\boldsymbol{E}^{(0)}$. Then, the adaptive procedure would continuously update the body-cursor mapping E^t as a function of the evolving user's movements.

As a result of the tuning, the a-AE updated the network using a history-batch of the last 60s of the IMU data. 10 steps of BGD with a learning rate $\alpha = 10^{-4}$ were applied to the network trained in the previous iteration.

Protocol

The protocol consisted of a reaching task. The participant sat in front of a 24in LCD computer screen, positioned about 1m away at eye level. The current position of the cursor and the targets was displayed on the screen as circles of 0.8 cm and 2.4 cm diameter respectively, similarly to previous studies involving a 2D reaching task (Farshchiansadegh et al., 2014). Participants were asked to move the cursor over the targets as rapidly as possible. The sequence of target presentations was the same for all the participants. A target was considered reached after the cursor had been kept inside the target for at least 250 ms. The chosen time allowed discriminating between movements actually directed to and ending in the target and movements that brushed by the target. The protocol, as summarized in Fig. 4.3, included two phases of reaching: training and test.

Training phase

The training phase was divided in 8 epochs. Within an epoch, the participants had to reach 4 targets (Fig. 3, grey dots) 8 times each. After completing a reaching movement, participants had to move the cursor towards the next target of the training sequence. The order of target presentation was pseudorandomized, with the condition that each target was not presented again before all 4 targets had been reached. In the last repetition of the 4 targets visual feedback of the cursor was removed, and the participants were asked to stop moving when they believed to be in the target (blind trials). The goal of these blind trials was to establish if the participants were guided by error feedback or if, instead, they formed a feedforward command based on an internal representation of the cursor space. During the first training epoch, the encoder for the adaptive group (A) was initialized with $\boldsymbol{E}^{(0)}$ and kept constant for 60s (baseline). After that, the map was iteratively updated as described in Table 6.1. Update of the map was suspended during blind trials.

Test phase

The participants practiced a center-out reaching task to eight target locations uniformly distributed on a circle (Fig. 4.3, blue targets). After each successful reaching, they were asked to move the cursor back to the central HOME target (Fig. 4.3, green target). In each

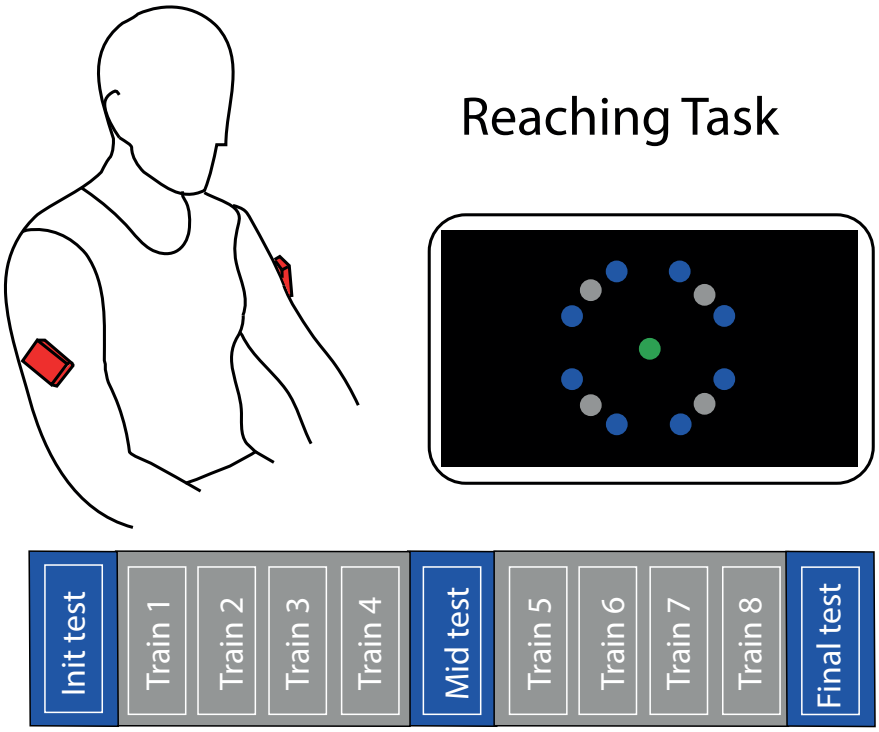

Figure 4.3 Setup for the reaching task and training protocol. The participant was sitting in front of a computer and was controlling a cursor using signals generated by IMUs (red boxes). Training (grey) and test (blue) targets were uniformly distributed on a circle. Four training targets were placed in four directions ($45° + k90°$), while eight test targets were placed in eight directions ($22.5° + k45°$). Each target was placed at the same distance L of 10.5 cm from HOME target.

test the eight targets were presented once. A total of three tests were presented: an initial test as baseline before starting the experiment, a midway test after four epochs of training, and a final test at the end of the experiment. For the whole duration of each test, the control group (F) practiced with $\boldsymbol{E}^{(0)}$, while the map update was suspended for the adaptive group (A). Hence both groups practiced with the same map during the initial test epoch.

4.3.5 Outcome measures

Sensitivity tuning

To evaluate the convergence and the stability of the a-AE in the offline sensitivity tuning we computed three metrics.

First, we quantified the tracking performance of the a-AE based on the **reconstruction error** (**RE**). RE is defined as the loss function described in Eq. 4.2., that was minimized every two seconds during the online retraining of the AE. A reduction of the reconstruction error implies that the mapping was providing a stable representation of the AE training data set.

To quantify the stability of the interface when using the a-AE, we considered the actual cursor vector $p^{(n)}$, and the cursor vector, $\hat{p}^{(n)}$, that would have been obtained without updating the encoder:

$$\begin{aligned} \hat{p}^{(n)} &= E^{(n-1)}(q^{(n)}) \\ p^{(n)} &= E^{(n)}(q^{(n)}) \end{aligned} \tag{4.7}$$

where $E^{(n-1)}$ and $E^{(n)}$ are the two time-consecutive encoders and $q^{(n)}$ is the movement set used in the training of the AE at the current update iteration. Note that, even though the a-AE at consecutive update iterations ($E^{(n-1)}$ and $E^{(n)}$) was trained with different movement set q, here we computed the cursor trajectory using the same set $q^{(n)}$. This allowed us to evaluate the **static jump**, defined as the L_2 norm between the final k value of $\hat{p}^{(n)}$ (end of an update iteration) and the initial value of $p^{(n)}$ (beginning of the next update iteration):

$$static\ jump = \sqrt{(\hat{p}_k^{(n)} - p_1^{(n)})^2} \tag{4.8}$$

Finally, we defined the **rate of changes** between consecutive updates by computing the L_2 norm between $\hat{p}^{(n)}$ and $p^{(n)}$:

$$rate\,of\,change = \sqrt{\sum_{i=1}^{k} (\hat{p}_i^{(n)} - p_i^{(n)})^2} \tag{4.9}$$

Online test

Four different measures were chosen to evaluate and compare the performance of the fixed and the adaptive group during the online test of the a-AE. Two of these were explicitly associated with the task requirements:

- **Trials completed** (**TC**): number of reaching trials completed during the training phase within a specific time frame. We computed the number of trials completed during baseline and then during time intervals of two minutes, including the final two minutes of training. Blind trials were not considered in this metric.
- **Endpoint Error** (**EE**): the Euclidean distance between the target position and the cursor position at the time of the cursor re-appearance during blind trials.

The other indicators measured performance not explicitly associated with the task requirements:

- **Linearity Index** (**LI**): maximum lateral deviation from the straight line connecting the beginning and end of cursor movement divided by distance between the same points. This is an index of straightness of cursor movements.
- **Movement Smoothness** (**MS**): number of peaks in the cursor velocity profile. We considered every peak larger than a threshold that was set to be 15% of the maximum speed of each trajectory. This is a measure of trajectory smoothness.

Besides analyzing the task-related performances, we also wanted to examine how the parameters of the a-AE evolved in time as well as to compare the strategies of the fixed and the adaptive group. Three metrics were devised for such purposes.

To assess the tracking ability of the a-AE, we monitored the a-AE reconstruction error at each update iteration of online adaptation of the interface, as we did in the offline tuning. From this measure, we derived the **Variance Accounted For** (**VAF**), defined as:

$$VAF = \left(1 - \frac{var(q - \hat{q})}{var(q)}\right) * 100 \tag{4.10}$$

Higher values of VAF were associated with an increased goodness of fit of the AE at the n update iteration, and consequently a sign of an enhanced tracking ability of the a-AE towards participant's latest movements. We then compared the VAF of the adaptive and the fixed group. In case of the a-AE successfully tracking user's movement manifold, we expected its VAF to be consistently higher than that of the fixed algorithm. Both groups were presented with a non-linear encoder (Eq. 4.4). We made the hypothesis that participants learned to minimize the "wasted" motion in the articulation space of the q's. To characterize how the control strategy (or, equivalently, the movement distribution) was changing throughout training, we computed the **bi-dimensionality index**. It was defined as the VAF by an AE model trained over each of the eight training epochs. The AE architecture was the same as in Fig 4.1. Finally, we wanted to determine whether the AE encoders in the a-AE converged to a specific structure. This required evaluating the similarity between different AE structures. The problem of assessing the similarity between neural networks had been widely studied (Kornblith et al., 2019; Morcos et al., 2018). Here, we followed a procedure described in (Raghu et al., 2017) that makes use of the Canonical Correlation Analysis (CCA) (Thompson, 2005). Specifically, we used the CCA to compare the final encoder structure obtained at the end of the training between any couple of participants (i, j). The inputs of the CCA were:

$$\begin{aligned} p_i^{t_{end}} &= E_i^{t_{end}}(x) \\ p_j^{t_{end}} &= E_j^{t_{end}}(x) \end{aligned} \tag{4.11}$$

x was designed as a vector of 3000 samples drawn from an 8D gaussian distribution. To resemble the distribution of a typical participant, the mean and covariance of x were set equal to those of the movement data from the same participant we used during the offline tuning. Note that, by using the same synthetic input x, we ensured this metric to depend only on the structure of the encoder E. Since the CCA works by maximizing the correlation ρ^m between its m inputs, we determined the **representational similarity** as:

$$similarity = \frac{1}{2}\sum_{m=1}^{2}\rho^m \tag{4.12}$$

By maximizing the correlation between its inputs, the CCA ensures the representational similarity metric to be invariant to any affine transformation. Higher values of similarity implied that participants converged towards a unique encoder. As an example of the effect of the similarity between final AE structures in terms of cursor control, we applied to the IMU data recorded by a participant during the final test epoch the final encoders of two other

participants, one with a high and one with a low similarity index respectively (the resulting cursor trajectories are shown in the result section).

4.3.6 Statistical analysis

To test whether the adaptive control affected the number of trials completed per unit time with respect to the fixed control, we performed a repeated measures analysis of variance (rANOVA) with time (level 1: baseline, level 2: last 2 min of training epoch) as within-subjects factor and group (level 1: adaptive, level 2: fixed) as between-subjects factor. We were interested in testing the group per time interaction, and the effect of the group. We did not test for the effect of time alone, as the duration of level 1 and level 2 was different. Similarly, to test the effect of time and group on the other indicators related to kinematic performance during training we ran a rANOVA with time (1-2: first and eighth training epochs) as within-subjects factor and group (1-2: adaptive, fixed) as between-subjects factor. We verified that the assumption of rANOVA were met by testing the sphericity of the data with the Mauchly's test and the normality of the data with the Anderson-Darling test. As a result, all data were normally distributed and spherical. Post-hoc analysis (Bonferroni correction) was carried out to verify statistically significant differences among factors whose effect was found to be significant. The threshold for significance was set at 0.05. For the post-hoc analysis, the Bonferroni-corrected threshold of significance was set to 0.025 (0.05/2). All analyses were performed in Statistica (Statsoft, Tulsa, OK, USA).

4.4 Results

4.4.1 Offline sensitivity tuning of the adaptive AE

Fig. 4.4A shows the reconstruction error during the a-AE sensitivity tuning for different values of learning rate α and memory batch size k, together with the R^2_{adj} and the set point of convergence (5**tau*) derived from the exponential fitting. The smallest value of the learning rate ($\alpha = 10^{-5}$) prevented the algorithm to converge within an acceptable time-frame, as the set point of convergence was consistently above 4000s for every size of k. Using a short-time memory batch increased uncertainty in the values of the reconstruction error over the training epochs (Fig. 4.4A, grey lines). As a result, the exponential fitting yielded the lowest R^2_{adj} for all the values of α tested. With a bigger memory batch and a learning rate $\alpha = 10^{-3}$, the adaptive algorithm fit user's movements in the fastest time frame (19s and 23s with a mid-term and a long-term history batch respectively). The learning rate of 10^{-4} had

intermediate performance, allowing a significantly faster convergence than that of 10^{-5}, and slightly slower than 10^{-3}.

Fig. 4.4B shows the amplitude of cursor position discontinuity at the instant of the AE parameters' update. The higher the learning rate, the bigger was the jump of the cursor. Vice versa, the bigger the size of the history batch, the smaller was the jump. A similar trend was found with the rate of changes of the AE map (Fig. 4.4C). Namely, the encoder had larger changes with larger learning rates and smaller history-batch sizes.

4.4.2 Online test of the adaptive AE

The adaptive group completed a significantly higher number of trials per unit time than the fixed group (rANOVA group effect: $F(1,18) = 6.35, p = 0.02$, Fig. 4.5A). Specifically, we found a significant interaction between group and time (rANOVA group x time effect: $F(4,72) = 5.5, p = 0.03$). At the beginning of the training, in the first 60s when the map was equal for all participants (baseline) both the adaptive and the fixed group completed the same number of trials (post-hoc comparison baseline $p = 1$). However, immediately after, the group that started working with the adaptive AE increased the rate of trials completed, with respect to the group that continued working with the initial map. Performance differences increased over time and were found to be significant in the final two minutes of training (post-hoc comparison last 2min: $p = 0.009$).

Both groups improved their accuracy over training time ($F(1,18) = 24.6, p < 0.001$). With the adaptive algorithm, participants outperformed the fixed group in terms of accuracy when reaching the targets during the blind trials (Fig. 4.5B). However, this difference was not found statistically significant (rANOVA group effect: $F(1,18) = 2.5, p = 0.13$).

Finally, with practice all participants moved the cursor along straighter lines (rANOVA time effect:$F(1,18) = 70.97, p < 0.001$, Fig. 4.5C) and with increasing smoothness (rANOVA time effect:$F(1,18) = 116.8, p < 0.001$), Fig. 4.5D). At the beginning of the training phase, initial performance of the fixed and the adaptive group were comparable for each metric. Performance differences became more evident during the second epoch of training, even though not significantly.

Participants from both groups learned to organize their movements in a latent manifold that progressively evolved toward a bi-dimensional structure across training (Fig. 4.6A). In fact, the bi-dimensionality index increased with training for both groups and eventually coincided at the end of the training. Noticeably, we recorded a significant difference when looking at the Variance Accounted For by the AE map over the training duration. The fixed

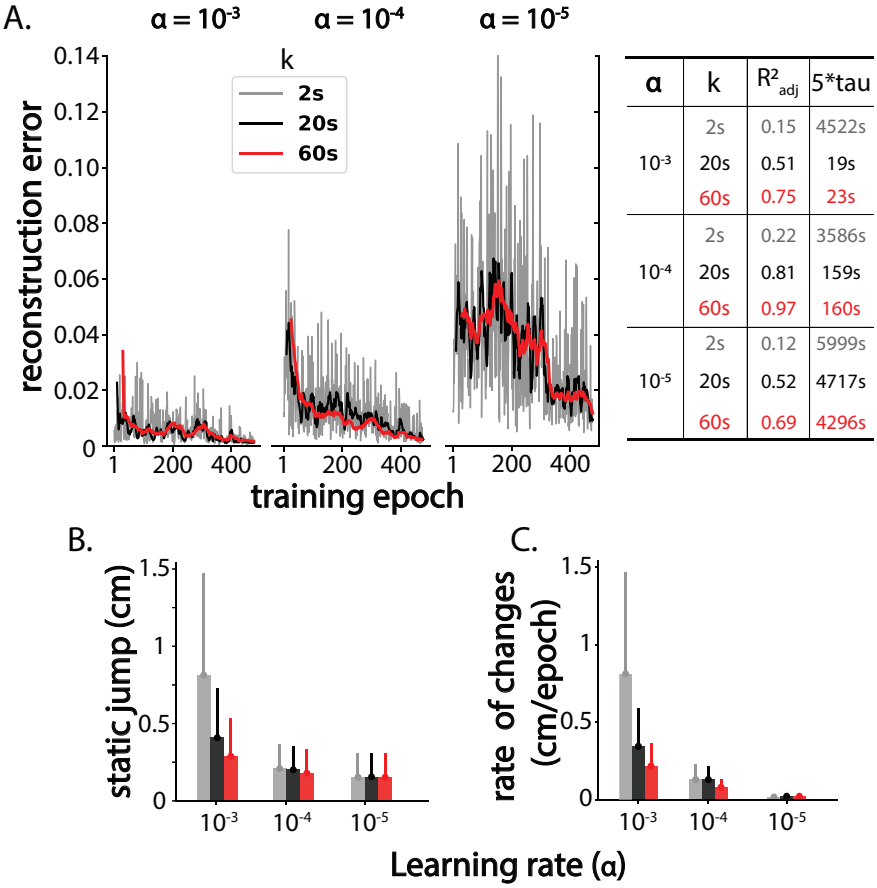

Figure 4.4 Summary of sensitivity tuning results. Panel A: reconstruction values with $\alpha = 10^{-3}, 10^{-4}, 10^{-5}$ and three different values of history-batch size (2s – grey, 20s – black and 60s – red). Note that, on the x-axis, one training epoch is equal to 2s. A table containing the adjusted coefficient of determination R^2_{adj} and the set point of convergence 5**tau* after fitting a double exponential is shown for each combination of hyperparameters. Panel B: mean and standard error of static jump values during training epochs. Panel C: mean and standard error representing the a-AE rate of change during training epochs. The same color scheme was used in every panel.

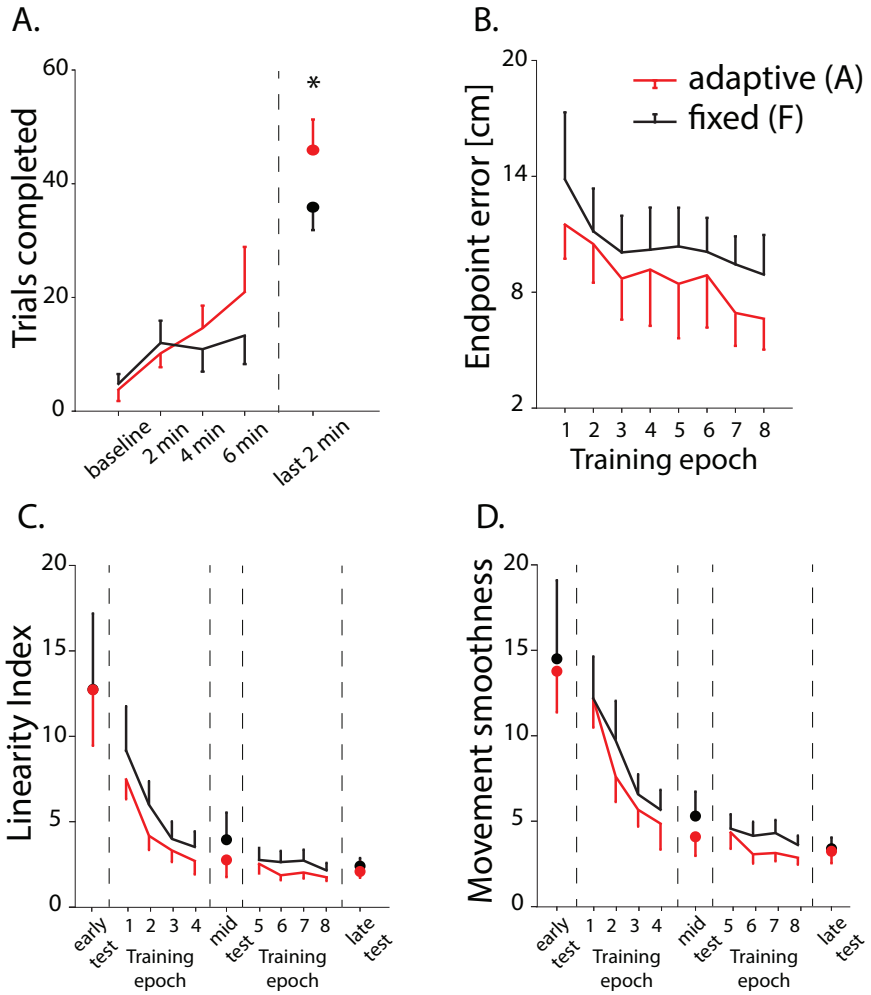

Figure 4.5 Performance metrics for adaptive (red) and fixed (black) group. Panel A: Number of trials completed during baseline, after two, four, six minutes following baseline and during the final two minutes of training. The asterisk represents a significant difference between groups during the last 2 min of training. Panel B: Endpoint error during blind trials of each training epoch. Panel C-D: Linearity index and movement smoothness values during each training and test epoch. Mean values across participants are plotted with 95% confidence interval.

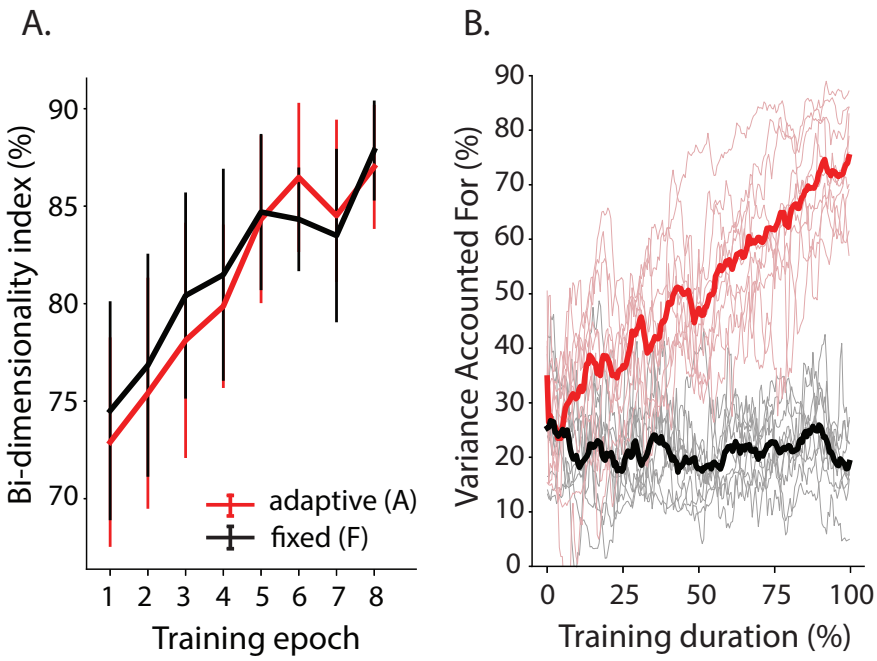

Figure 4.6 Panel A: Bi-dimensionality index values for adaptive (red) and static (black) group during each training epoch. Mean values across participants are plotted with 95% confidence interval. Panel B: Variance accounted for (VAF) values for adaptive (red) and static (black) group during each batch of the training duration. Mean across subjects is shown as a bold line for both groups.

map (Fig. 4.6B, black lines) was not able to capture the movement manifold of its users, as the values of VAF plateaued at around 25% for the entire duration of the reaching task. On the other hand, allowing the AE map to change over time consistently increased the fitness of the a-AE to the user's movement manifold (Fig. 4.6B, red lines) and eventually reached the peak of 80% VAF at the end of the training duration.

Fig. 4.7 shows the similarity matrix representing the similarity between the encoders obtained at the end of the training phase for the adaptive group. Participants were ranked according to the inter-subject similarity between final AE structure. Six participants converged towards fairly comparable encoders (S1, S2, S4, S6, S8, S10 - 6x6 matrix in the top left corner, Fig. 4.7), while four did not (S3, S5, S7, S9).

An example of how such similarity might have impacted the control of the cursor in shown in Fig. 4.8. Namely, if we were to substitute the final AE map that a participant used in the late test phase with that of another participant whose map had a comparable structure, the same participant would have been able to cover the target space much more consistently (Fig. 4.8A) than with a map whose similarity was not as pronounced (Fig. 4.8B).

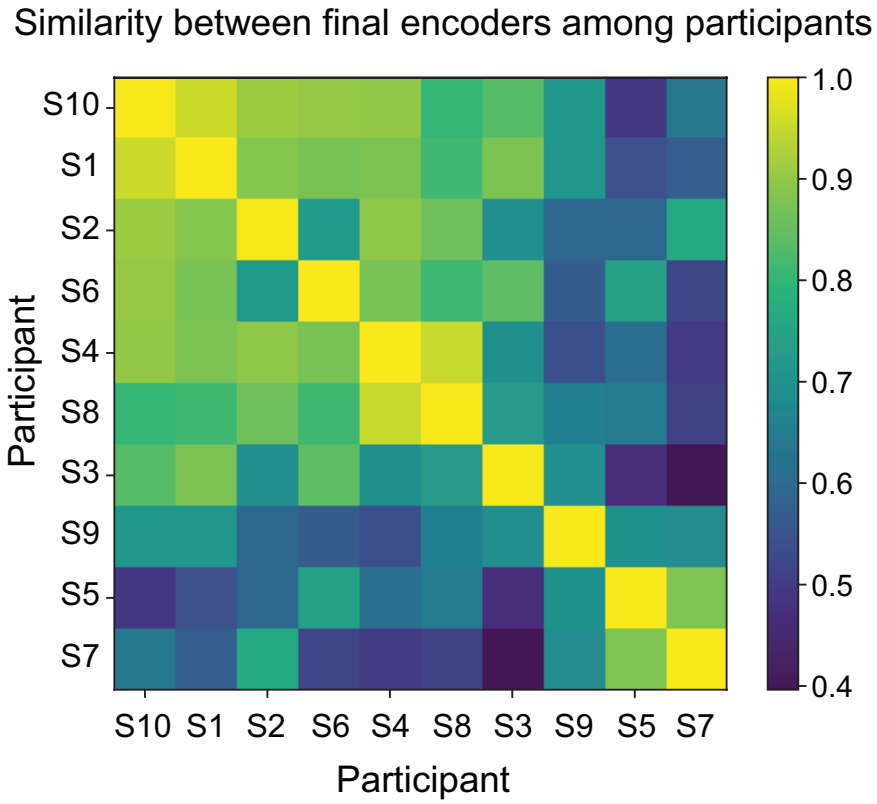

Figure 4.7 Similarity matrix representing the similarity between final encoders among participants. The higher the value of a cell, the more similar the final encoders of the two participants of that cell. The matrix has been ranked to cluster participants with similar final encoders.

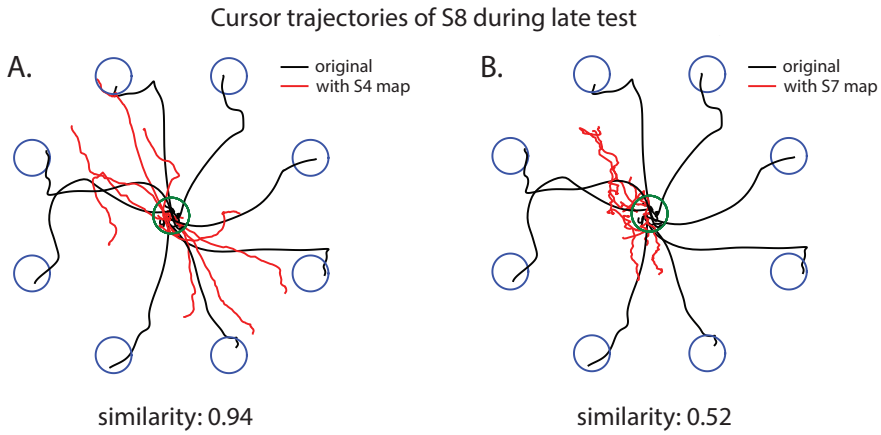

Figure 4.8 Cursor trajectories of one participant (S8, black lines) during the final test epoch. We took the IMU data of S8 recorded during the final test epoch and applied the final encoder of a participant with whom S8 had a high similarity (S4, red lines, Panel A) and a low similarity (S7, red lines, Panel B).

4.5 Discussion

In this study we proposed an adaptive platform based on the use of an iterative non-linear autoencoder to implement unsupervised tracking of user's manifold for improving the ease-of-use of a human-machine interface. First and foremost, our results support the use of a non-linear AE as a proficient control map within the body machine interface scheme. Moreover, the adaptive approach led to an increased representational efficiency of the interface decoder while concurrently increasing users' task-related performance, both in terms of number of trials completed over time and accuracy during reaching of blind trials. This result suggests that the online co-adaptation process encourages the development of a more accurate internal model. Importantly, the proposed approach has three salient features that makes it appealing in many applications other than the one tested here: i) it cancels the cost of interrupting the operation of the device to perform decoder recalibration, ii) as no information about the state of the task and/or intended task goals is needed, the manifold tracking algorithm can be applied to a great variety of contexts, iii) it does not rely on the existence of a stable neural or movement manifold to compensate for decoder instabilities, allowing it to be applied in the earliest stages of interface operation, when the formation of new neural strategies is still on-going.

4.5.1 Autoencoder networks can proficiently control low-dimensional devices

There is an increasing enthusiasm about using autoencoder networks in the field of human-machine interfaces. AEs give the freedom of choosing the type of network architecture and level of complexity (*e.g.*, linear/non-linear, generative/convolutional/recurrent, number of hidden layers and neurons per layer), thus potentially allowing to better match the degree of complexity of the input signal and hence allow for a great variety of applications. For instance, AE-based approaches have recently been employed for adversarial and variational domain adaptation (Farshchian et al., 2018; Hsu et al., 2017) and for extracting precise estimates of neural dynamics (Pandarinath et al., 2018b). However, applicabilty of non-linear AEs for the control of low dimensional devices have seen limited efforts (Vujaklija et al., 2018).

The use of AEs, or more specifically of their encoder sub-network, as a forward map in a BoMI has been first proposed by (Pierella et al., 2018). In that study, an AE encoder was used to map muscle activities recorded via EMG to the coordinates of a computer cursor, similarly to the case considered here. However, there is a crucial difference between these

two implementations. In (Pierella et al., 2018), the AE cost function was modified by adding a constraint to Eq. 4.2 to force the latent space dimensions to follow a hierarchy in terms of their variance, mimicking PCA (hierarchical autoencoder (Scholz et al., 2008)). The inclusion of this constraint may limit the reconstruction power of the autoencoder. Hence, the approach is not optimal for accurately estimating the latent space defined by BoMI user movements, with potential repercussions on the ease-of-use of the interface. In order to avoid this potential pitfall and maximize the variance accounted for by the latent representation (Portnova-Fahreeva et al., 2020), here we decided to use a "vanilla" autoencoder that performs DR by minimizing Eq.4.2 without any added constraint. The AE architecture used in our study was very parsimonious, since the addition of more parameters did not increase the variance explained on the movement calibration data set. Even with our parsimonious choice, the AE in Fig. 4.1 allowed retaining 93% of the variance of the calibration data set in its latent encoding, while a linear DR algorithm (PCA) would have accounted only for 72% of the original variance on the same data set.

Few other studies have explored the representational power of AEs in HMIs. In the work of (Vujaklija et al., 2018), an AE was used to map the EMG activity of forearm muscles during instructed wrist movements with one degree of freedom (DoF) (*i.e.*, flexion/extension, or radial/ulnar deviation) into cursor displacement along a line. (Losey et al., 2020) used an AE to extract non-linear motion primitives from guided movements of a robotic arm. The users controlled the robot by selecting latent actions with a joystick and the AE decoder sub-network mapped the selected action back into the high DoF robot movement.

However, while the use of non-linear AEs is ideal when modelling any process that has a non-linear structure (such as body movements, or neuronal activity), their use in the closed-loop schema of a BMI (or, equivalently, a BoMI) might result challenging. On the one hand, having a better estimate of the BoMI user's intentions might allow the user to generate more accurate motor commands. On the other, non-linearity comes with the loss of a major advantage of linear control - the possibility to obtain a full repertoire of actions from the direct summation of simpler actions. Nevertheless, this study found that learning to operate a non-linear interface is not only possible, but comparable to what has been observed with a linear interface (De Santis and Mussa-Ivaldi, 2020), thus providing a proof of concept for the use of non-linear autoencoders for human-machine interaction.

In this study, we also took a step forward and utilized non-linear AEs in an iterative way for closed-loop adaptation with a learning agent. The online update of AEs posed different challenges, such as the hyperparameter tuning that we discussed in section §4.2. Nevertheless, we showed that iteratively updating an autoencoder used as control map to

track user's latent manifold is not only successful (Fig. 4.6B) but led to an increase in user's task-related performance when operating the interface.

4.5.2 Map adaptation converges during offline test

When dealing with any learning algorithm, the common problem of choosing a set of optimal parameters is typically referred to as (hyper)parameter tuning (Claesen and De Moor, 2015). In our case, there were three main parameters that required such tuning: the learning rate α, the number of steps β of BGD and the size of the history-batch. Each parameter influenced the a-AE tracking capability of user's movement manifold.

First and foremost, we had to define which time frame of user's previous movements could be considered representative of its movement distribution. In this sense, the advantage of using a short-term memory is that the update operation would be the least computationally expensive. However, the a-AE memory needs to contain a sufficiently comprehensive description of user's movement statistics to achieve the desired convergence rate target. Hence, a short-term memory is the most prone to being locally biased especially in the initial phase of practice with the interface, as the movement variability may be confined to a small region of space. This may prevent the adaptive algorithm from converging within an acceptable time frame (see 5**tau* in Fig. 4.4A). Thus, one would want to consider a memory whose size is increased (*i.e.*, a mid-term or long-term memory). Increasing the memory batch expands the time horizon of user movements accounted for by the a-AE, making the online training more representative of user's evolving distribution and less prone to overfit (or, equivalently, smoother), at the expense of a computationally more expensive update. The choice of one over the other was motivated by looking at the concurrent value of learning rate.

Results showed that a learning rate of $\alpha = 10^{-5}$ prevented the algorithm from converging within an acceptable time frame. Therefore, the a-AE would not have been able to keep track of the evolving distribution of user's movements. With $\alpha = 10^{-3}$, the adaptive algorithm fit user's movements in the fastest time frame (25s with a long-term history batch of 60s, Fig. 4.4A). Ideally, the adaptive algorithm should wait for the user to develop a meaningful motor strategy before converging. However, during the first stages of learning, BoMI users typically complete only one reaching trial within the time frame of 20/30s. As mentioned by (Müller et al., 2017), when the machine learns too fast, the co-adaptive process is unstable and unable to converge. The additional metrics that we computed seem to support this claim, as the rate of changes (Fig. 4.4B) and the static jump (Fig. 4.4C) increased proportionally

with the network learning rate. On the other hand, with $\alpha = 10^{-4}$ (and a long-term memory of 60s), the algorithm managed to reach the set point of convergence within 160s, which was considerably closer to the target value of 2 min.

As a result, we opted for the intermediate learning rate ($\alpha = 10^{-4}$) and a long-term history batch (k=60s). This choice was further motivated by the work of (Dangi et al., 2013; Orsborn et al., 2011, 2012), recommending time scale of updates of 1 minute for applications to adaptive brain-machine interface, and those of (Danziger et al., 2009; Davidson and Wolpert, 2003; Golub et al., 2018; Orban de Xivry and Lefèvre, 2015) showing that interfaces that changes too abruptly during online operation are detrimental for users' learning.

We ran the sensitivity tuning with a prefixed (and arbitrary) value of $\beta = 10$. Since this value allowed to satisfy our tuning goals, we decided to exclude this parameter from the tuning procedure and chose it as our final parameter for the online reaching test.

4.5.3 Map adaptation leads to superior performance

With the proposed online adaptive algorithm, participants significantly outperformed the fixed group in terms of number of trials completed over time. Indeed, by the end of the training, the adaptive group reached a significantly higher number of targets than the fixed group (Fig. 4.5A). It is interesting to notice that, while both groups unsurprisingly started with the same level of performance, the contribution of the a-AE started improving the performance of the adaptive group already after four minutes. The performance difference between the two groups became increasingly significant over time, finally reaching its peak in the last two minutes of training. These results suggest how, after an initial period of co-adjustments between the BoMI users and the interface, the adaptation of the latter towards user's movement manifold successfully helped participant's ease-of-use of the interface. Moreover, we found how the a-AE promoted the development of an internal representation of the cursor space, as participants were able to identify the position of targets more precisely during the blind trials (Fig. 4.5B, with a nearly approached significance threshold). This result is consistent with that obtained by our previous study of an online co-adaptation with a linear interface (De Santis and Mussa-Ivaldi, 2020). Interestingly, both the fixed and the adaptive group organized their movements towards a structure that gradually resembled that of a two-dimensional manifold as they learned how to operate the interface (Fig. 4.6A). This confirms our hypothesis that participants, through motor learning, distributed their motions so as to match the dimensionality of the sensory feedback, in line with previous studies (Mosier et al., 2005; Ranganathan et al., 2014).

Remarkably, the proposed adaptive algorithm was successful in tracking and tailoring its user's movements (Fig. 4.6B). This further validates the choice of the a-AE parameters made after the sensitivity tuning and, together with the reported improvement in task-related performance, confirms our initial hypothesis that human-machine interaction can be promoted if the interface low dimensional output and the human's manifold are isomorphic. The use of an adaptive interface that tracked participants' manifold over time allowed shaping their movement distribution in a manifold resembling that of the interface itself. As a result, we speculate that the Adaptive group converged to a more efficient inverse model of the AE encoder. The finding that the Adaptive group was less dependent on visual feedback with respect to the Fixed group supports this idea. We believe that the Fixed group was still able to efficiently operate the interface even in the presence of more inaccurate estimates of the inverse AE encoder because of the redundant nature of our interface. Namely, participants could have used different motor strategies to complete the task. Therefore, learning of an accurate predictive component was not necessary as long as participants could have relied on the visual feedback to compensate for aiming inaccuracies. Moreover, the Static group could have also attained high accuracy from the prolonged exposure to the same map.

4.5.4 Map adaptation is guided by the individual learning trajectory

If on the one hand the a-AE was successful in tracking the movement manifold of each participant, on the other we noted that the changes in the encoder were not consistent across participants. Namely, we could divide the BoMI users into two main groups: those who converged towards a similar encoder structure at the end of the training (S1, S2, S4, S6, S8, S10 - Fig. 4.7) and those whose maps did not converge to a particular similarity value (S3, S5, S7, S9 - Fig. 4.7). All the participants were able to reach a satisfactory level of performance regardless of whether the map was converging towards a particular structure. Furthermore, as mentioned before, in addition to being adaptive, the redundant character of the interface allowed each participant to complete the task with a different strategy. We speculate that the final encoder was merely a result of the learning trajectory of each participant. Indeed, if two participants were to exhibit the same movement dynamics (*i.e.*, strategy) from the start to the end of the practice, it is reasonable to assume that they would have converged towards the same encoder. Since this was not always the case, we concluded that some participants effectively learned the task in a different way. It is important to remark that the a-AE was able to guide them to an efficient resolution of the task regardless of the inverse

model employed by the participant. This is a remarkable characteristic that increases the generalization capability of the proposed interface.

4.5.5 Perspective on current adaptive interfaces

The problem of building adaptive interfaces is raising increasing interest in the realm of human machine interactions. Here we focused on the development of an adaptive interface that provides a seamless interaction with the user during online operation of the interface. In the field of BMIs, the closed-loop adaptation of the interface is driven by some policies associated with the user's known movement intention (Vidaurre et al., 2011). This results in supervised adaptation as, for example, it would require the user to perform pre-selected movements to guide the update of the interface parameters (Dangi et al., 2014; Orsborn et al., 2012). Another possible approach to adapt the interface so as to account for user's motor strategies could stem from Reinforcement Learning (RL) (Sutton and Barto, 2018), where a software agent continually interacts with an environment and take actions in order to maximize some reward. Previous studies (DiGiovanna et al., 2008; Mahmoudi et al., 2008; Sanchez et al., 2009) have used RL-inspired algorithms to modify the agent's (the interface) behaviour according to what was considered desirable for the user. The RL approach, however, still requires the definition of a value function in order to assign a reward to an observed action.

To improve the generalizability of the adaptive interface across tasks, an unsupervised approach is most appropriate. If on the one hand Mehring et al. proved that an unsupervised co-adaptation is theoretically possible (Gürel and Mehring, 2012), on the other there is still a clear gap of knowledge in the implementation of an unsupervised adaptation concurrent with the operation of the interface. In this sense, a recent work by Yu et al. proved that an unsupervised re-alignment of intrinsic manifold of neural activities could stabilize interface performance in the presence of recording instabilities (Degenhart et al., 2020). There are similarities between the approach described in our study and theirs, as both share the basic rationale of exploiting latent manifolds for updating the interface. However, unlike Yu et al., here we are concerned with tracking a time-varying manifold, rather than stabilizing it against artefactual changes. Therefore, our co-adaptation approach is intended to operate immediately, without waiting for the users having developed a stable motor strategy.

We want to stress that the this study represents a novel approach for designing an adaptive interface, as it avoids constraints that other state-of-the-art interfaces commonly have, such as the need of interrupting the operation of the device, of relying on information about the

state of the task, or relying on the existence of a stable neural or movement manifold. As such, directly comparing the performance of our interface against those of other interfaces with completely different nature is not straightforward. Perhaps a comparison can be made with an earlier study performed by our own group (De Santis and Mussa-Ivaldi, 2020), in which a linear model, iterative PCA, was used to align the space of body movements with the space encoded by the interface. However, the experimental setups differed in the placement of the inertial sensors (two IMUs on the arms in this study, four IMUs – two on the arms and two on the forearms – in the previous study). Nonetheless, the study presented in this manuscript confirmed some of the findings of our previous one, in particular the ability of the adaptive interface to lead to the development of a more faithful internal representation of the control problem (Fig. 6B). Furthermore, differently from the linear case, we found that, with the proposed non-linear adaptive interface, participants were able to significantly outperform those practicing with a static map in terms of target acquisition rate. We believe that this difference might be due to the superiority of non-linear models in estimating non-linear latent manifolds (Tenenbaum et al., 2000) such as those spanned by upper body kinematics. This has further implications for enhancing the generalizability of the interface proposed in this study, as it might be more appropriate when dealing with manifolds derived from processes such as neural recordings, as well as when developing a controller for multi-degrees-of-freedom devices.

4.5.6 Limitations

The choice of the learning rate α of the network was motivated by the tuning procedure we ran as a first stage of this study. However, our implementation of the online co-adaptation did not consider the rate of the BoMI user learning the task. In this sense, an online co-adaptation between the learning rate α of the network and the learning rate of the user operating the BoMI might improve the ease-of-use of the interface itself. This could be achieved by implementing a policy that adapts α accordingly to the user's learning time scale. A higher value of α might be beneficial during the first part of the learning curve (cognitive stage (Fitts and Posner, 1967)). During this stage of learning, humans typically present high-movement, and consequently, high-performance variability. As stated before, however, there is a dangerous threshold, after which forcing people to explore more will, most likely, prevent them from learning the task at all. We did not find any study that deals with the problem of developing an algorithm that accounts for this threshold, and, as a result, further efforts in this sense are required. Going back to the policy design, as soon as the policy registers an

increase in user's performance, the value of α could be decreased to encourage a strategy consolidation. The policy might eventually set the learning rate of the network to a minimum value when user's performance is about to plateau. Our study, however, suggested that letting the interface continuously adapting, even after the user reached a sufficient capability of controlling the cursor, did not disrupt motor learning. In other words, the proposed interface found its own stability without any supervision. We assume that this was a result of the design choice to consider an extensive portion of user's latest movements. Thus, as long as the statistical distribution of those movements was not changing, the encoder was not changing substantially, de facto autoregulating itself.

Chapter 5

Controlling an assistive robotic manipulator with a non-linear body machine interface

5.1 Introduction

The consequences of a neurological disorder vary with its nature and severity. For example, after spinal cord injury (SCI), people present damage at the efferent and afferent neural pathways that lead to a broad structural and functional reorganization of the nervous system. A complete injury, classified as ASIA 'A' (Roberts et al., 2017) usually results in a total loss of function, both motor and sensory, below its level. Individuals with an incomplete SCI typically maintain some function below the injury level (ASIA 'B' to 'E'), with the sensory function that is better preserved than the motor function (van Middendorp et al., 2011). Depending on the level and severity of the lesion it is also possible to predict the likelihood of developing hyperreflexia which might ultimately lead to spasticity (Adams and Hicks, 2005; Dietz, 2000; Lance, 1980). For example, 93% of individuals with cervical SCI classified as ASIA' A' are likely to develop spasticity. This is also the case for 78% of subjects classified as ASIA' B' to' D' (Maynard et al., 1997). Among the consequences of spasticity, there are muscle weakness and an increase of muscles' undesired activations

Preliminary results of the content of this chapter have been published as Giordano, M.*, Rizzoglio, F.*, Mussa-Ivaldi, F.A. and Casadio, M., 2020. Controlling an assistive robotic manipulator with a non-linear Body-Machine Interface. In *International Conference on NeuroRehabilitation*. Springer, Cham.

during voluntary contractions (Fouad and Tetzlaff, 2012). These conditions affect the ability to perform selective movements and coordinate body motions, and ultimately limit the performance of functional activities of daily living (ADLs).

Common daily activities, such as walking, grooming and object manipulation, could be facilitated through the use of assistive devices, such as powered wheelchairs and robotic manipulators. However, designing interfaces for controlling such devices can be challenging – especially when dealing with devices with many degrees of freedom (DoFs). For instance, the use of a traditional interface, such as a hand-joystick for controlling a powered wheelchair, does not take into account the limited arm and hand mobility or coordination of individuals with cervical SCI, thus potentially disrupting the operation of the device (Fehr et al., 2000). Alternative controllers, such as Sip-and-Puff, head-mounted switches and tongue-based devices can be used, altough they provide the user with only a limited set of discrete commands (Bastos-Filho et al., 2013; Koike et al., 2016; Struijk et al., 2017).

Here, we propose an approach based on the framework of body machine interfaces (BoMIs) (Casadio et al., 2012). BoMIs exploit the fact that, even after a severe injury, many individuals retain some movement, especially of their head and shoulders, that can be used to control external devices. BoMIs convert high-dimensional body signals (*e.g.* upper body kinematics, muscle activities) into lower-dimensional, latent, commands to operate the device. As a result, BoMIs allow individuals with motor disabilities to overcome some of their impairments. Importantly, BoMIs allow to recover continuous control of external devices, as opposite for example to methods based on the recognition of discrete gestures (Geethanjali, 2016).

The use of BoMIs has been tested in situations involving the control of a computer cursor (Rizzoglio et al., 2020), a powered wheelchair (Thorp et al., 2015), a robotic manipulator (Jain et al., 2015), and quadcopters (Miehlbradt et al., 2018). To do so, BoMIs typically rely on a linear dimensionality reduction (DR) technique - Principal Component Analysis (PCA) (Wold et al., 1987). On the one hand, linear control comes with the advantage of obtaining a full repertoire of actions from the direct summation of simpler actions. Also, linear models typically have a low computational cost and are able to produce consistent results, given their deterministic nature. On the other hand, linear models cannot account for non-linearities of the input dataset, thus overestimating the dimensionality of the input signals (Tenenbaum et al., 2000) and potentially disrupting the operation of the external device. Moreover, despite its widespread use, PCA might not be suited for controlling devices with a large number of DoFs, as the variance explained by its latent dimensions significantly drops after the first couple of them. Here we propose the use of non-linear autoencoder (AE) networks to

overcome these difficulties. AEs are artificial neural networks that compress data into a latent code via non-linear DR. Not only non-linear AEs are able to estimate the dimensionality of their input more parsimoniously than PCA (Portnova-Fahreeva et al., 2020), but their structure can also be customized to obtain a variance that is more uniformly distributed across its latent dimensions. Although there are only a few examples of the use of AEs within a BoMI framework (Pierella et al., 2018; Vujaklija et al., 2018), the hypothesis of this study is that the customizability offered by AEs can make them an ideal candidate for operating a device with many DoFs. To test our hypothesis, we developed and tested a BoMI based on non-linear AEs that allowed users to control a 4D virtual robotic manipulator.

5.2 Methods

5.2.1 Experimental apparatus

The BoMI recorded an 8D kinematic signal from the participants' upper arms and transformed it to a 4D control signal to operate the virtual robotic manipulator. The kinematic signal was recorded by two Inertial Measurement Units (IMUs, *BN0055 Sensors, Bosch, Gerlingen-Schillerhfiohe, Germany*) positioned bilaterally on participants' arms. An on-board Kalman filter allowed the acquisition of each sensor orientation in the quaternion format. Data from the IMUs were acquired in real-time using a WiPy 2.0 (*Pycom, Guildford*). The software of the WiPy was developed with MicroPython.

The kinematic data were then used to control a virtual robot simulated on Gazebo (Koenig and Howard, 2004). In this study, we simulated the MICO robotic arm (*Kinova Robotics, Canada*), a modular robotic manipulator that is suitable for advanced assistive manipulation research and can easily be installed anywhere, like on powered wheelchairs. The arm can safely handle common objects for daily use by utilizing the two under-actuated fingers. The control of the virtual robot was achieved via ROS packages (Quigley et al., 2009).

We used an AE to map the 8D vector q of body movements - quaternions from the IMUs - into the 4D vector p of angular movements of the joints of the virtual MICO. An AE is composed of two parts: an encoder that compresses the input into a lower-dimensional latent representation, or code, followed by a decoder that converts the latent representation into the output, with the same dimensionality as the input. In the BoMI context, the code layer can be used to generate the control signals for the external device. For instance, when controlling a 2D computer cursor, an AE with 2 code units (CUs) would allow mapping the first CU to control cursor movements along the x-axis, and the second CU to control movements along

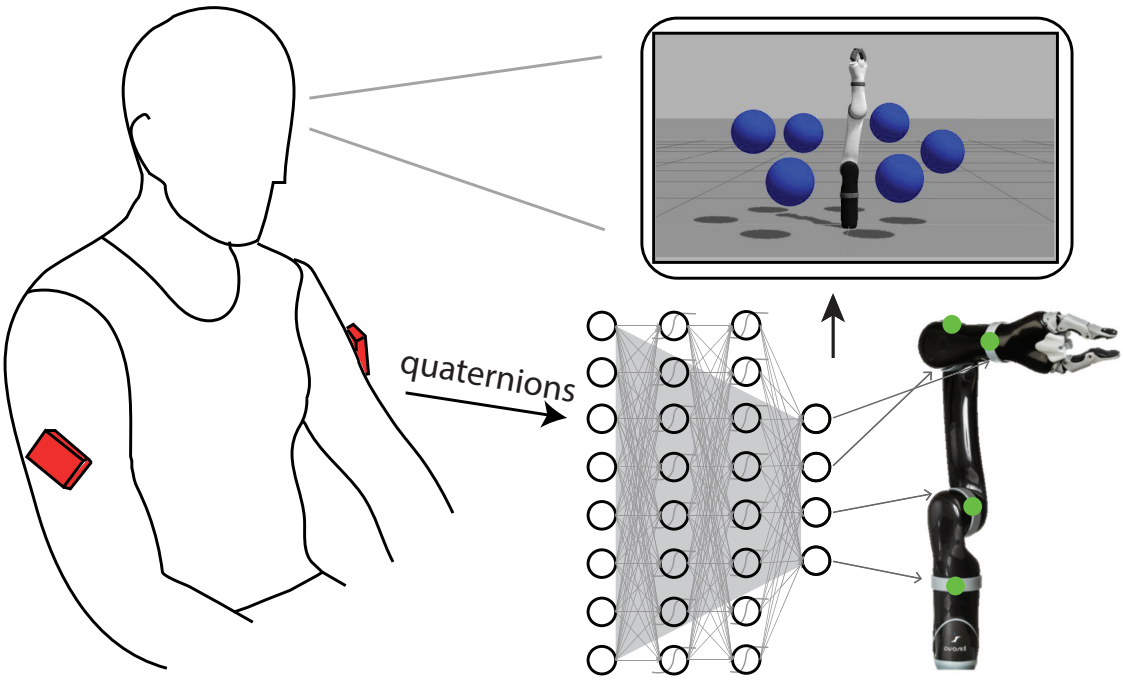

Figure 5.1 Setup of the experiment. Quaternions from two IMUs (red boxes) were mapped via the AE encoder to the four joints (green dots) of the virtual MICO. The participant received visual feedback on a PC monitor.

the y-axis. In this study, the external device controlled via the AE had 4 DoFs. Therefore, the BoMI forward map that transformed the high-dimensional (8D) body-space signals into the lower-dimensional (4D) control-space was set to be the encoder subnetwork of an AE with a 4D CU layer. Specifically, we mapped the first code of the AE to control the movements of the wrist angle of the manipulator – the most distal joint - and proceeding in order toward the base so that the fourth code mapped the robot's base-joint angle, as illustrated in Fig. 5.1. Each code was re-scaled to fit the maximum extension of each joint angle (360° for the base-joint angle, 180° for the others).

The encoder network was initialized following a validation procedure whose details are presented in the next section.

5.2.2 Validation of the AE

The validation procedure aimed at selecting the parameters of a 4D AE that, when taking the 8D kinematic body-signal as input, would yield (i) minimum information loss and (ii) uniform distribution of variance across the CUs.

The validation was performed using a dataset obtained via a calibration procedure in which a subject (age 25, male) that did not participate in the following study was asked to freely move his arms for 60s. The movements performed were intended to explore the range of possible arm movements without executing uncomfortable or extreme gestures. The calibration procedure shared the same experimental setup considered in this study – two IMUs were placed bilaterally on the participant's arms to record the 8D body signal. The kinematic data recorded during calibration had a total of 3000 samples (60s with sampling frequency 50 Hz). The calibration dataset was split into 80% of training points (2400) and 20% of test points (600).

The parameters and architecture of the AE needed to be fine tuned for designing the most ideal forward map for the online control of the MICO. The optimal AE network was chosen among two candidate architectures:

- *under-parametrized AE*: first, we trained an AE with a smaller number of parameters (364) w.r.t. the number of training samples (2400). The network was designed with four hidden layers (two in the encoder subnetwork and two in the decoder subnetwork), with eight neurons per layer.

- *equally-parametrized AE*: then, we trained an AE with a higher number of parameters (2652), that were roughly equal the number of training samples (2400). This network also had four hidden layers, but 30 neurons per layer.

Each of the two architectures consisted of a nonlinear activation function, a hyperbolic tangent (*tanh*, (Kalman and Kwasny, 1992)), for each hidden layer and a linear one for the code and output layers. The initial AE weights were chosen following the Xavier normal initialization (Glorot and Bengio, 2010), while the initial biases were set to zero.

In order to measure the information preserved by each AE during training, we computed the **Variance Accounted For** (**VAF**), defined as the percentage of variance of the calibration body-signal explained by the AE:

$$VAF = \left(1 - \frac{var(q_c - \hat{q}_c)}{var(q_c)}\right) * 100 \tag{5.1}$$

where q_c is the original calibration dataset and $\hat{q}_c$ is the dataset reconstructed by the AE. A VAF of 100% indicated that the dimensionality of the calibration dataset equaled four.

Then, we quantified how the latent variance (v_i) was distributed across the i^{th} AE CU as the percentage with respect to the overall variance of all considered CUs, and defined it as **latent variance**, v_{lat_i}:

$$v_{lat_i} = \frac{v_i}{\sum_{i=1}^{4} v_i} \tag{5.2}$$

AEs are non-deterministic models (unlike PCA). Therefore, the VAF and latent variance might differ depending on the initial conditions of the AE (refer to Chapter 2 of this thesis for a more detailed explanation). While the VAF is typically less prone to vary depending on the initial condition, the latent variance can change drastically. Therefore, to evaluate the impact of different initializations on the VAF and the latent variance, we trained each type of network with five different initial parameters.

The goal of the validation was to select the AE architecture that would yield a high VAF (*e.g.*, $VAF > 90\%$) concurrently with a uniformly distributed variance across the code units (*e.g.*, $v_{lat_i} \sim 25\%$ for each CU). In addition to the various AEs, we also trained PCA on the same training dataset and compared its performance with that of nonlinear AEs.

The results from the tuning procedure are obtained considering the test points only and are presented in section 5.3.1. The final choice leaned towards one of the five under-parametrized networks. Therefore, we assigned the encoder sub-network derived after training the selected AE structure on the training dataset as the BoMI forward map for the online control of the MICO. The transformations operated by the encoder to obtain the 4D control vector p from the 8D body vector q were:

$$\begin{aligned} layer_1 &= tanh(w_1 * q + b_1) \\ layer_2 &= tanh(w_2 * layer_1 + b_2) \\ p &= w3 * layer_2 + b_3 \end{aligned} \tag{5.3}$$

where w_j were the weights and b_j, $j = 1:3$ were the biases of the encoder subnetwork.

5.2.3 Participants

For testing the online control of the virtual MICO, we enrolled 12 unimpaired participants (5 females and 7 males, age 24.25 ± 2.75). They did not have any evidence or known history of postural, muscoloskeletal or neurological disorders, and exhibited normal joint range of motion and muscle strength. They signed a consent form approved by the institutional review board (n. registro CE DIBRIS: 009/2020). All the participants were assigned with the same BoMI forward map, derived from the validation procedure.

5.2.4 Experimental protocol

The experimental protocol consisted of a 3D center-out reaching task. Participants performed the task while comfortably seated. An LCD computer screen, positioned in front of the participants about 1.5 m away at eye level, was used to display the virtual manipulator's current position and the targets. The experiment was divided into two identical sessions performed on different days of the same week. The maximum duration of a single session was set to two hours. Participants were asked to reach six small balls (targets) placed uniformly along a circle, at different heights, with the robot's EE. A trial was considered successful if the target remained between the fingers of the manipulator's gripper for 500 ms.

The starting point of each reaching movement was always the same and corresponded to the robot configuration resulting from mapping a resting position in the body motion space via the AE. Participants were instructed to reach the robot's initial configuration by returning every time to the same resting position, with both hands on their tights. If they had difficulties reaching the initial configuration, we re-tared the IMUs to bring the manipulator back in the correct configuration while asking them to be in the resting position. This was necessary to correct for offsets or drifts in the orientation estimate over time. The protocol included two phases of reaching: training and test, with the following order of presentation:

- *initial test*: a first test of ability in the reaching task (also called baseline).
- *initial training*: first training with the interface.
- *mid test*: a second test of participants' performance in the middle of the session.
- *final training*: final training with the interface.
- *final test*: a final test of participants' performance at the end of the training.

If the participant failed to complete the whole protocol within 2 hours, the training phase would be interrupted before its completion and the final test would still be performed. Following are the details of each phase:

Training phase

This was the longest part of the experiment. Participants had to start from the resting position, and then they were required to reach a target with a timeout of 60 seconds. If they failed to complete the trial within 60 s, the target would disappear, and they had to come back to the initial position before the following target was presented. There were six different targets to

be reached. They were equally distributed in a circle around the base of the manipulator, with a radius of 9 cm and at different heights. Both the initial and the final training were divided into four blocks, each composed by 18 reaching trials, for a total of 72+72 = 144 targets.

Test phase

The purpose of this phase was to assess the performance of each participant at different times during the session. Starting from the resting position, the participants had to reach the presented target, with a timeout of 20 seconds. If they failed to do so, the target disappeared, and the trial was considered unsuccessful. At the end of each reaching, regardless of it being successful or not, participants were asked to come back to the resting position and get ready for the next target. During each of the three tests (initial, mid, and final), users had to reach 18 targets, for a total of 18+18+18 = 54 targets. Targets were equally distributed in a circle around the manipulator base in six different positions, rotated by 30° and at different heights with respect to the training phase. During test, we also increased the distance from the center of the manipulator to the base of each target and set it to 12 cm.

5.2.5 Outcome metrics

Different variables were chosen to evaluate the performances of the participants during the online test of the BoMI.

- **Reaching Time** (**RT**): time from the first appearance of a new target to when the target has been reached successfully. A reduction of RT indicates improvement in performance.
- **Success Rate** (**SR**): percentage of trials completed within timeout during test blocks.
- **Normalized path length** (**NPL**): distance travelled by the robot end effector to reach the target divided by the nominal distance. This is an index of straightness of end effector movements.
- **Smoothness index** (**SI**): number of peaks in the end effector speed profile. We considered every peak larger than a threshold set to be 20% of the maximum speed of each trajectory. This parameter is a smoothness index.

Beside performance indices, we evaluated how users distributed their movements over training time and which strategies they employed in order to move the robot efficiently toward the targets.

To address the first point, we studied the movement distribution both in the 8D IMU space and in the 4D robot joint space. We computed the**Variance Accounted For** (**VAF**), defined as the percentage of variance explained by a DR method during training epochs, as specified by Eq. 5.1. Specifically, we trained a linear (PCA) and a nonlinear (AE) DR model - both 2D and 3D - on movements from both spaces recorded during 12 consecutive reaching trials. An increase of VAF indicates that movements were converging towards a dimensionality specified by the latent dimensionality of the DR model. Here, we also wanted to compare linear and nonlinear models to assess the differences between them when estimating the dimensionality of a kinematic signal. We expected the nonlinear method to account for more variance than the linear one.

Training PCA also allowed us to investigate which strategies participants employed during the online operation of the robot - specifically how they organized the movements of each robot joint over training time. We considered the first 3 principal components (PCs) derived from training PCA on the 4D robot joints kinematic data and evaluated their **loadings**. Since we computed the loadings on the 4D robot joint space, we obtained four loadings for each PC - one for each joint. The higher the loading of a joint, the higher its contribution to the related PC. We computed the loadings across day 1 and day 2 to highlight potential differences in the motor strategies employed by participants across days. In this case, we only considered data the reaching movements (both successful and unsuccessful) during the training phases.

Finally, we computed the variance (**VAR**) of each of the four robot joints recorded during training over 12 consecutive reaching trials to determine which of the four robot joints contributed the most to the overall movement variance of the robot. Moreover, a reduction of VAR indicates more consistent joints movements over training time.

5.2.6 Statistical analysis

To test the effect of time on the indicators related to performance during training, we ran repeated measures analysis of variance (rANOVA) with training time (1-24: start training day 1 to end training day 2, each level considered 12 consecutive trials) as within-subjects factor. Post-hoc analysis (Fishers LSD test) was carried out to verify statistically significant differences among levels, in case the training time was found to be significant. The threshold for significance was set at 0.05. All analyses were performed in Statistica (Statsoft, Tulsa, OK, USA).

5.3 Results

5.3.1 Validation of the AE

A latent dimensionality of four allowed to retain over 95% of the variance of the calibration dataset with both PCA and nonlinear AE (Fig. 5.2A). The number of AE parameters did not impact the VAF, as the difference between the under-parametrized and the equally-parametrized AEs was negligible. As expected, the latent variance changed dramatically depending on the initialization of the AEs, as indicated by the high standard deviation in Fig. 5.2C. However, one of the initializations of the under-parametrized AE yielded a nearly uniform distribution of variance across the codeunits (Fig. 5.2B). As a result, we chose the parameters of this network (specifically of its encoder sub-network) for the online test of the BoMI. Note that, if we were to use PCA instead, we would end up with a control space in which the variance dropped significantly for the last principal components (PCs). This would have a dramatic impact during the online test, as the movement of the robot joint associated with PC3 and PC4 would be extremely limited. A common practice for PCA-based BoMI is to normalize the variance of each PC, so as to make each robot joint variance uniform. However, PC3 and PC4 accounted for only 6% and 4% of the overall calibration movement variance. Thus, they had a significantly smaller signal-to-noise ratio than PC1 and PC2, and augmenting their gain would probably lead to a less efficient control of the associated robot joints.

5.3.2 Task-related metrics

With training, participants were able to move the robot significantly faster towards the targets ($F(23,253) = 96.489, p < 0.001$, Fig. 5.3A). The biggest performance improvement was recorded during the first day of training ($p < 0.001$) while performance within day 2 slowly continued to improve ($p < 0.001$). Interestingly, participants were able to retain their performance across days, as the difference in RT between the end of day 1 and the start of day 2 was not significant ($p = 0.103$). Similarly, there was an effect of training time on both the straightness ($F(23,253) = 36.399, p < 0.001$, Fig. 5.3B) and the smoothness ($F(23,253) = 26.404, p < 0.001$, Fig. 5.3C) of robot trajectories. After the first day of training, they had the greatest performance improvement ($p < 0.001$ for both NPL and SI). In the second day of training, movement straightness was retained ($p = 0.938$), while movement smoothness was not ($p = 0.04$). Additional training during the second day allowed

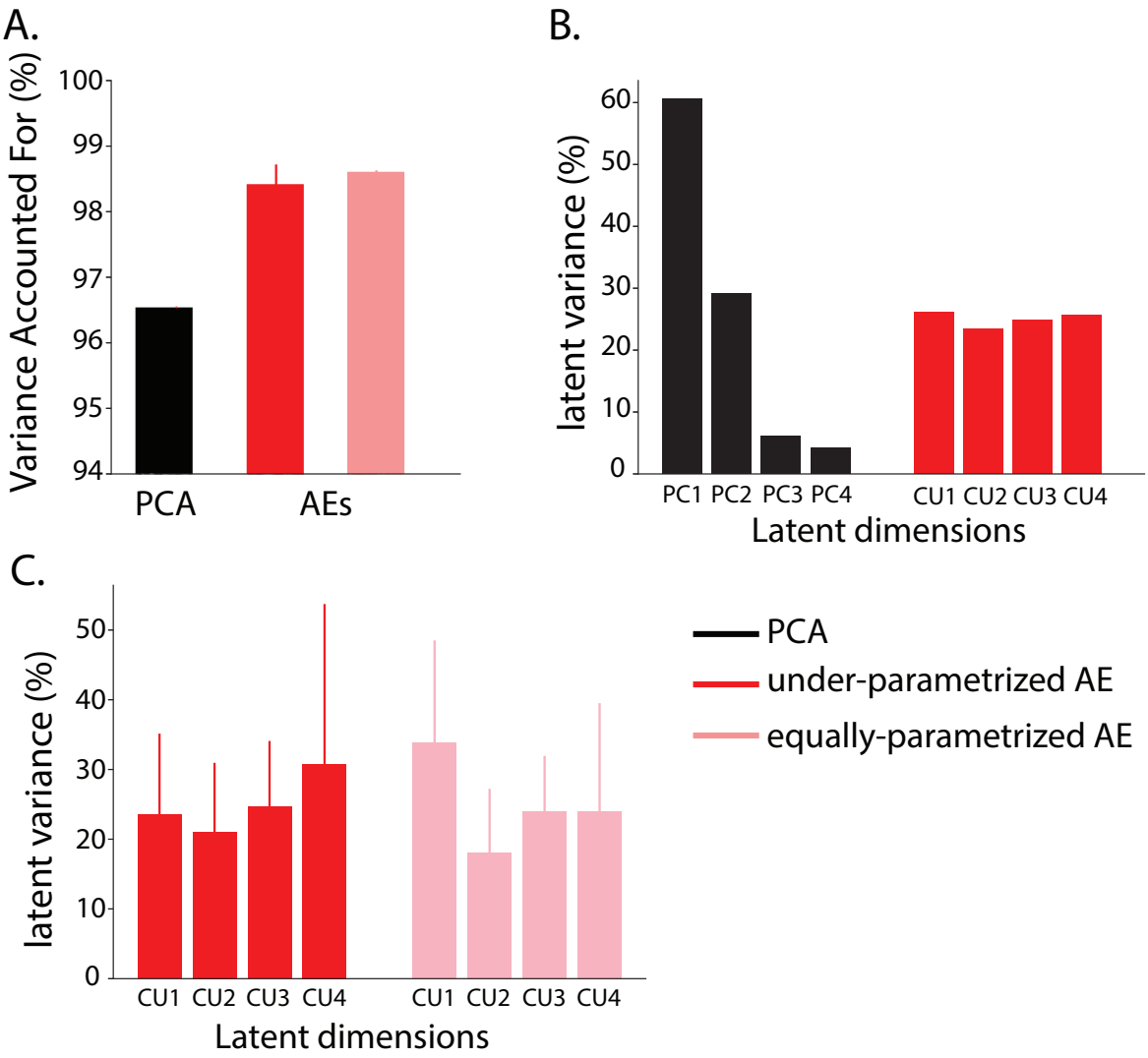

Figure 5.2 Panel A: Variance Accounted For by PCA (black bar), under-parametrized AE (red bar) and equally-parameterized AE (pink bar) on the calibration dataset. Mean and standard deviation across five different initializations are shown for the nonlinear AEs. Panel B: Latent variance for PCA (black) and the selected under-parametrized AE (red bars). Nonlinear AE allowed to distribute the variance uniformly along its latent dimensions, while PCA presented the typical drop of variance for the last PCs. Panel C: Mean and standard deviation across the five different initializations for the under-parametrized AE (red) and the equally-parametrized AE (pink).

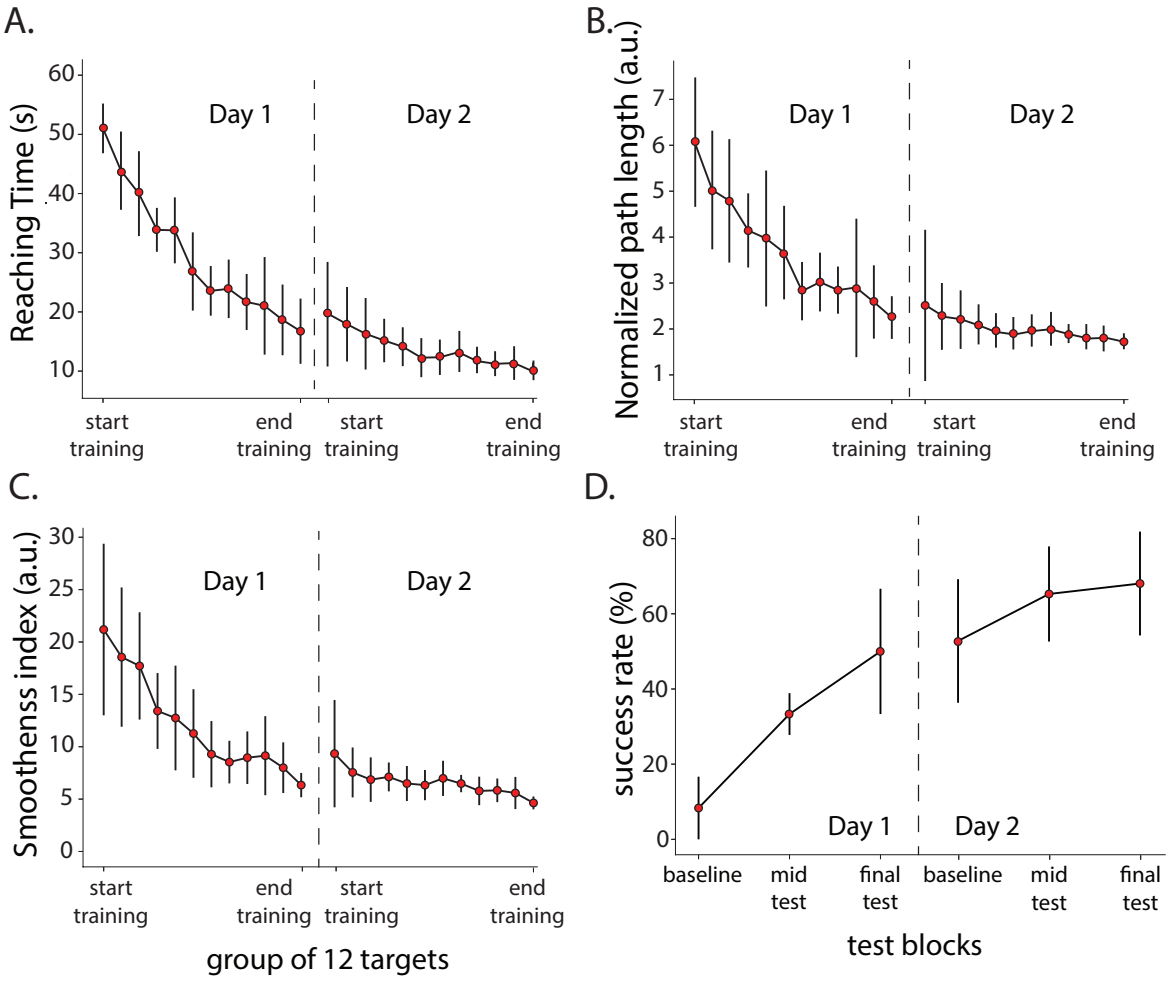

Figure 5.3 Reaching time (Panel A), normalized path length (Panel B) and smoothness index (Panel C) over training blocks, divided into sets of 12 trials. Panel D: Percentage of successful reaching movements over the different test blocks. Mean and standard deviation across participants is plotted for each set. A dashed line separates values across day 1 and 2.

participants to furtherly improve (NPL: $p < 0.001$, SI: $p = 0.008$) and converge towards a stable level by the end of the session.

During the test phase, there was a significant increase in the number of trials completed within 20s as a result of training ($F(5,55) = 47.834, p < 0.001$, 5.3D). The biggest improvement was recorded during the first day of training, with participants completing little over 50% of the trials at the end of the session ($p < 0.001$). Importantly, this level of performance was retained at the start of the second day ($p = 0.341$). Participants managed to complete a significantly higher number of trials at the end of day 2, with almost 70% of them resulting successful ($p < 0.001$).

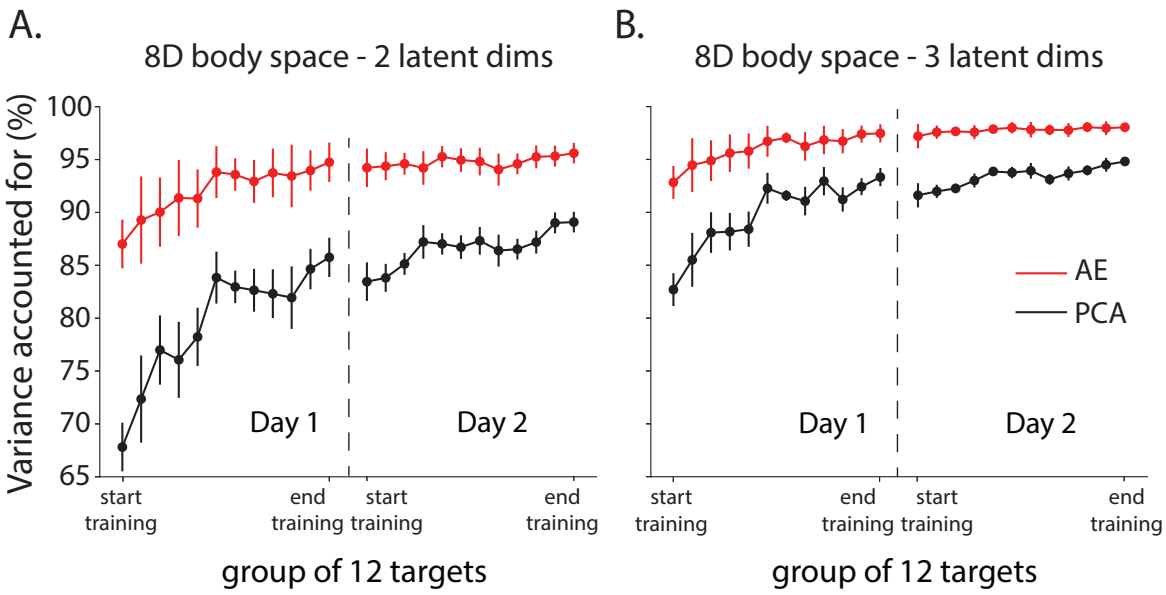

Figure 5.4 2D (Panel A) and 3D (Panel B) VAF calculated on 8D IMUs space with nonlinear AE (red) and PCA (black). Mean and standard deviation across participants is shown for sets of 12 consecutive reaching trials.

5.3.3 Movement strategy metrics

Figure 5.4 shows the VAF calculated on the movements of participants' arms as recorded by the IMUs, both with a 2D and a 3D non-linear AE and PCA. At the beginning of the first day, participants started exploring which movements were more appropriate for reaching the targets, as indicated by the lower values of VAF (both 2D and 3D). After few training epochs, they slowly learned how to move the robot more efficiently, as the values of VAF started to increase. Interestingly, the VAF difference between the end of day 1 and the start of day 2 was negligible. Finally, by the end of the second day of training, it is safe to assume that participants learned to distribute their movements in a 3D space, as the VAF of a 3D non-linear AE reached its maximum value of over 95%. Moreover, a nonlinear AE was able to estimate the dimensionality of body movements more accurately than PCA, as the latter VAF values were significantly lower than the former ones throughout the entire duration of the experiment.

Figure 5.5A and 5.5B shows the VAF calculated on the 4D movements of the robot joints, respectively with a 2D and 3D non-linear AE and PCA. In this space, the difference between VAF values of AE and PCA was less remarkable than that recorded in the body space. Moreover, it is clear that the dimensionality of the robot joint space was consistently 3D right from the start of training, as the 3D VAF values were immediately over 99%. Figure

5.5C shows the loadings distribution over each robot joint for the first 3 PCs during day 1 and day2. The loadings of PC1 and PC2 changed considerably during day 2, while those of PC3 resulted similar in both days of training. Interestingly, the standard deviation of all loadings during day 2 decreased drastically from that of day 1, indicating that participants moved the robot joints more consistently with each other after the first day of training.

Figure 5.6 shows the variance of the movements of the four robot joints. Participants were able to move the base joint (joint 1) and the EE joint (joint 4) much more efficiently during day 2, with respect to day 1, as indicated by a sharp decrease of VAR in day 2. Moreover, the standard deviation of the VAR was significantly lower during day 2 for all the robot joints. This indicates how, during day 2, movements of the robot joints were much more consistent across participants than during day 1.

5.4 Discussion

This study delivered three main findings: all participants (i) managed to acquire a high level of skill when operating the 4D robot as a consequence of training; (ii) retained the performance they had at the end of the first day of training; (iii) started moving the robot with individualized strategies but converged towards a single solution over the course of training. We consider these findings as supporting a future implementation of our interface as an assistive tool for people with motor impairments.

5.4.1 Controlling a device with multiple DoFs

Complex assistive devices, such as robotic manipulators, can return independence to people with motor impairments. However, controlling a device with multiple DoFs is a challenging task (Cowan et al., 2012; Kaliki et al., 2012). Commercially-available interfaces can be either proportional or discrete. With a discrete interface, users are given the control over a selection of "modes", where each mode controls a single DoF of the device. Thus, in order to fully control the device, users must manually switch between the modes. The main advantage of a discrete control is its simplicity, as a 1-dimensional interface is effectively used to control the n-dimensional device. The immediacy of use of such interfaces favored a wide use of applications, as in the case of powered wheelchairs (Bastos-Filho et al., 2013), and robot manipulators (Jain et al., 2015). However, discrete controllers do not allow their users to regulate the magnitude of the control signal, thus resulting far too simplistic to be considered ideal for performing dexterous tasks.

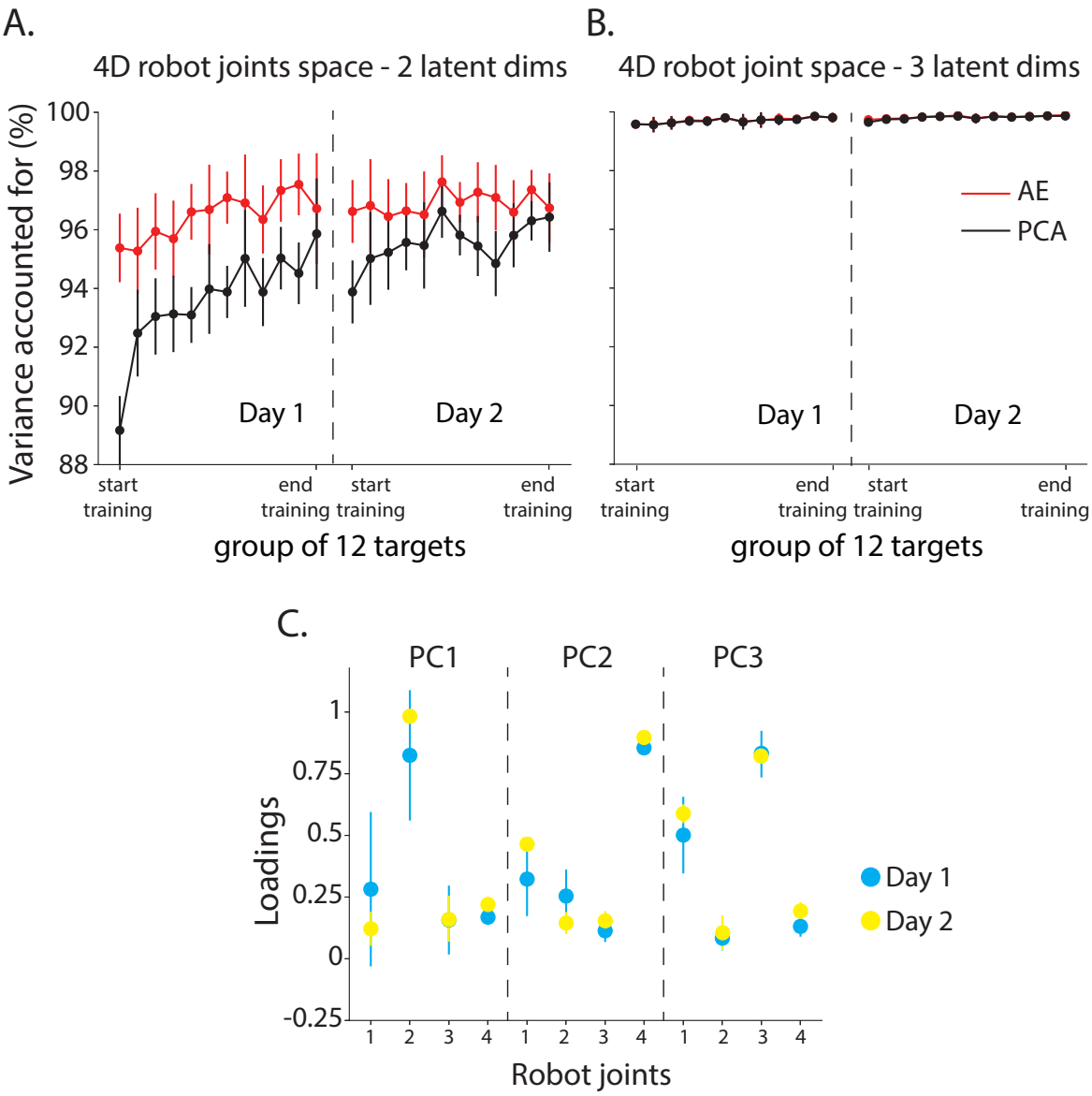

Figure 5.5 2D (Panel A) and 3D (Panel B) VAF calculated on 4D robot joints space with nonlinear AE (red) and PCA (black). Mean and standard deviation across participants is shown for sets of 12 consecutive reaching trials. Panel C: Loadings distribution over the robot joints for the first 3 PCs during day 1 (blue dots) and day 2 (yellow dots).

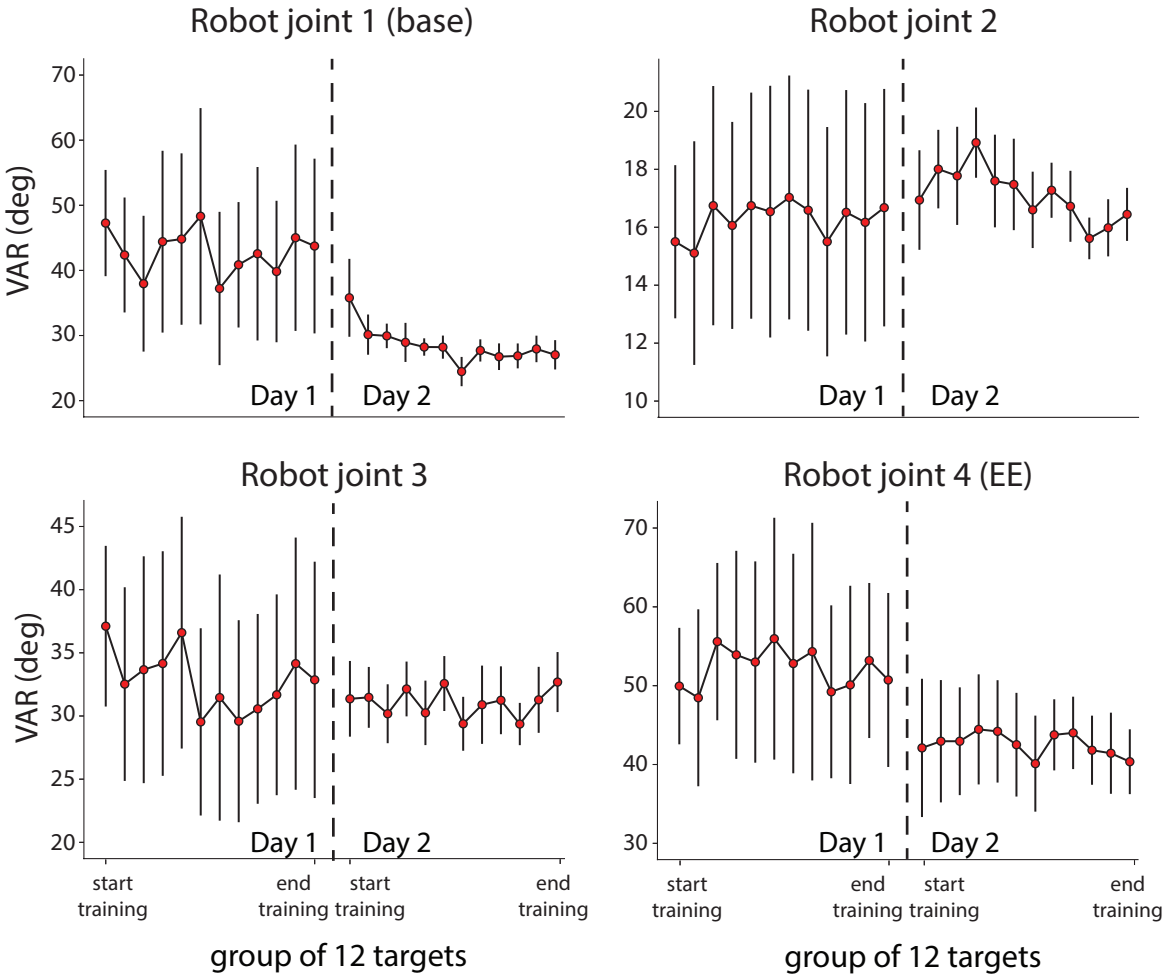

Figure 5.6 2D VAF calculated on 3D robot EE space with nonlinear AE (red) and PCA (black). Mean and standard deviation across participants is shown for sets of 12 consecutive reaching trials.

Therefore, one might want to consider a continuous controller. Nonetheless, adopting a continuous control comes with the possibility of introducing an excessive computational burden to the user. If the user were to share the control with the machine, this burden can be alleviated and the user autonomy over the device while still be maintained, even if not fully. However, as pointed out in a recent study by Javaremi and Argall (Javaremi and Argall, 2020), there is no one-size-fits-all method for control sharing, as each person is unique in their desired control preference.

In this sense, the approach used in this study stands out because, not only is continuous, but it also allows users to operate the robot in a fully autonomous way. Moreover, the unsupervised nature of the BoMI makes it ideal for applications in a clinical context, since it can be tailored to its user's residual movements. As far as this study is concerned, the computational burden of learning how to simultaneously control 4 DoFs was not excessive, as all participants managed to acquire a high level of skill by the end of training. However, future iterations should consider the implementation of an adaptive interface, as the process of co-adaptation has shown to guide the redundancy resolution towards increase movement efficiency (De Santis and Mussa-Ivaldi, 2020).

5.4.2 Linear vs. nonlinear DR in human-machine interfaces

The question of whether a linear or a nonlinear DR method should be used as a control map is of crucial importance when designing a human-machine interface. In this study, we tried to investigate the differences between these two approaches when estimating kinematics of the upper body and focused on the use of a nonlinear model to directly control an external device.

Linear DR models, such as PCA and Factor Analysis (FA), have seen the most adoptions in both brain (Degenhart et al., 2020; Santhanam et al., 2009) and body (Carrington et al., 2014; Farshchiansadegh et al., 2014; Rizzoglio et al., 2020) machine interfaces. Using a linear model is convenient due to its low computational cost and the possibility to obtain a full repertoire of actions via the summation of simpler actions. However, linear models are intrinsically limited when dealing with the problem of reducing the dimensionality of a nonlinear signal. Since most of the neurophysiological signals have a nonlinear nature, a linear algorithm would not be able to derive a low-dimensional, latent space correctly (Tenenbaum et al., 2000). Thus, since both BMI and BoMI typically rely on the use of such latent space for controlling an external device, the user experience with a linear interface could potentially be disrupted.

An alternative solution would be to exploit nonlinear DR methods. As demonstrated by a recent study, a nonlinear AE is able to accurately estimate the dimensionality of full hand kinematics (Portnova-Fahreeva et al., 2020). Our results seem to confirm this condition, as the VAF of each of the BoMI spaces considered (8D-IMU, 4D-robot joints, 3D-robot EE) resulted significantly higher in the case of a nonlinear AE than with PCA.

Nevertheless, applications of nonlinear models within the HMI context did not historically arouse the same enthusiasm. One of the reasons for this is that nonlinear models are quite cumbersome to interpret and often need to be specifically tuned to properly suit a certain application. For instance, in this study, finding the most suitable AE structure required validation (or equivalently fine-tuning), which is naturally a time-consuming procedure. As a result, this approach might not be ideal for those contexts where immediacy of use is preferable. On the other hand, if one is willing to sacrifice the conveniency of linear models (*e.g.*, due to them being of deterministic nature), the intrinsic customizability of nonlinear methods makes them attractive to be tailored to any scenario. For instance, if one wanted to select the most ideal platform for controlling a device with multiple DoFs, we demonstrated that a properly tuned nonlinear AE allows to uniformly distribute the variance among the control dimensions. This would not have been possible with PCA (Fig. 5.2), thus making non-linear AEs more desirable for this application. Moreover, even if BoMI users could not exploit the property of superposition given by a linear control, the online operation of the interface was not harmed, as all participants managed to acquire a high level of skill at the end of the training sessions.

5.4.3 Motor strategies employed by participants to operate the robot

When users learn to operate a machine via BoMI, it is crucial to determine how the learning process takes place. Here, we addressed this point by studying (i) how users distributed their movements over training time and (ii) which strategies they employed in order to move the robot efficiently toward the targets.

Previous studies suggested that learning to efficiently operate a device with a BoMI comes with a reorganization of users' movements towards a structure whose dimensionality matches that of the device itself (De Santis and Mussa-Ivaldi, 2020; Pierella et al., 2017a). This is not trivial, since the redundant nature of the BoMI forward map would allow participants to maneuver the device with a combination of movements that lies on a higher dimensional space, which might be different each session and/or task. In those studies, the dimensionality of the device matched that of the task. In this study, we expanded on this matter by introducing

a "double" redundancy. Indeed, not only the 4 DoFs manipulator was redundant w.r.t. the higher dimensional body space, but it was redundant w.r.t. the 3D reaching task as well. Therefore, not only there were different body configurations that would have resulted in the same manipulator configuration, but also different manipulator configuration that would have resulted in the same position and orientation of the robot end-effector. This double redundancy allowed us to establish whether participants developed motor strategies to match the 3D space of the task or the 4D control of the manipulator. Results show that it was the dimensionality of the task to be matched, as the variance of body signals accounted for by a 3D AE converged to over 98% at the end of the training sessions (Fig. 5.4B).

Note that the aforementioned studies defined the experimental redundancy based on the variance accounted for by the first few principal components derived from PCA. Since PCA is a linear algorithm approximating a participant's control strategy with a linear manifold, this approach cannot fully account for the variance of body signals associated with a nonlinear control strategy. Hence, in case participants were to employ such non-linear control strategy, the experimental redundancy could only be approximated by PCA, thus potentially leading to misleading interpretations. The limitation of using linear DR methods for this sort of analysis was confirmed by our results, since we showed that a non-linear AE had significantly higher VAF than its linear counterpart (red vs black lines, Fig. 5.4 and 5.5). The take home message here is that non-linear methods should be preferred when making claims about the dimensionality of a certain space.

Nonetheless, training PCA allowed us to further enlarge our views on the motor strategies employed by the participants during the two days of training. Being a linear DR technique, PCA can give an intuition about how participants distributed the movements of the robot joints over training time. Interestingly, participants started moving the robot with individualized strategies but converged towards a single solution over the course of training, as indicated by the sharp decrease of the std associated with each loading (5.5C).

When looking at each robot joint singularly, we can see that participants learned to move the base joint (joint 1) and the EE joint (joint 4) more consistently and efficiently than with the other two (Fig. 5.6). This could be due to joints 1 and 4 being the most external ones, which made them presumably the most influential for the task's objective. Also, the base joint presented the highest variability in both days. This was probably a result of it being the joint that, by construction, had the highest range of movements (360°, vs 180° for the others). Investigating the movement variance of the singular joints confirmed that participants started moving the robot with individualized strategies but converged towards a single solution over

the course of training, as indicated by the sharp decrease of the standard deviation of VAR during day 2 (also Fig.5.6).

Part III

Autoencoders as a mean to model user's motor learning

Preface

Numerous studies suggested that our brain, in order to generate actions designed to realize a desired task, or adapt to external forces, develops an internal representation of the task and of the geometrical/mechanical environment. In the case of a subject operating an external device using a BoMI, the learning process consists in inverting the redundant many-to-one BoMI forward mapping. In the case of a subject operating an external device using a linear BoMI, a recent study (Pierella et al., 2019) described how users gradually formed this inverse model throughout the course of training. The final goal of my project was to verify whether this behavior could still be observed in the case of nonlinear BoMIs. Moreover, I investigated whether AEs can be used to represent and estimate user motor learning during the operation of the BoMI. Specifically, I aimed at empirically estimating users' inverse model via training of the AE decoder sub-network.

Chapter 6

A nonlinear model of motor learning through practice with a body machine interface

6.1 Introduction

Numerous studies suggested that our brain, in order to generate actions designed to realize a desired task, or adapt to external forces, develops an internal representation, or internal model, of such tasks and forces (Conditt et al., 1997; Kawato, 1999; Krakauer et al., 1999; Shadmehr and Mussa-Ivaldi, 1994; Wolpert et al., 1998). The brain forms two types of internal models as the learning takes place: a (i) forward model (FM) that predicts the sensory consequences of motor commands, and an (ii) inverse model (IM) that generates successful commands from movement goals. (Jordan and Rumelhart, 1992) further investigated the roles of forward and inverse models by considering how the learning process can occur via the concurrent learning of FM and IM. Specifically, they introduced the concept of distal learning. Considering a skill-learning task (*e.g.* shooting a ball in a basket), the learners have to transform their intention (*i.e.*, making the basket) into motor actions (*e.g.*, muscle commands), which are transformed by the environment into distal outcomes (*i.e.*, making or not making the basket). In the distal learning approach, the learner starts by forming a FM of the transformation from action to distal outcomes, and it does so by exploring the outcomes associated with various actions. Then, once the FM has been partially learned, it can guide the learning of an IM that transforms the intention into the most appropriate action that, when composed with the environment, yield the desired distal outcome.

Here, we extend the distal learning approach to the learning of a novel map established by a body machine interface (BoMI) (Casadio et al., 2012; Pierella et al., 2019). BoMIs translate high-dimensional movements of the upper body into lower-dimensional movements of a machine (*e.g.*, a computer cursor that users guide to a set of target locations). In this case, the environment that transforms the motor actions (body movements) into the distal outcome (cursor position) is the many-to-one BoMI forward map, and the FM is a model of the BoMI forward map that predicts the outcome of user's action. After an initial learning of this FM, the BoMI user has to learn which motor actions allow them to realize the intention of moving the cursor onto the target. In other words, the user has lo learn the IM, represented as the inverse of the redundant many-to-one BoMI forward mapping. A recent study (Pierella et al., 2019) mathematically described how users, in the case of a linear BoMI, not only generate an IM of the control map, but that their IM converges as a first-order exponential toward a particular attractor state (*i.e.* a single solution).

In this study, we expand the study of IM in the case of nonlinear BoMIs by proposing a procedure for empirically estimating the IM developed during the online BoMI operation using an autoencoder (AE) network (Kramer, 1991). AEs are a class of artificial neural networks (ANNs) that can implement a non-linear BoMI by setting their encoder sub-network as the BoMI forward map. The encoder parameters are typically obtained by unsupervised training over a set of movements recorded during an initial calibration. While training an AE requires the concurrent optimization of its encoder and decoder sub-networks, the latter is not directly included in the BoMI. Here, we exploit the fact that the decoder, being an ANN, can serve the purpose of function approximator (Hornik et al., 1989) and thus can potentially be used to model user's learning. In addition, the encoder-decoder structure of an AE is trained so that the decoder approximate an inverse of the encoder map (that is, the BoMI forward map). Thus, the IM formed by the BoMI user can ideally be modeled as a network of the same architecture as that of the AE decoder.

Here we aim at using the AE decoder to build an empirical estimate of the evolving IM that predicts the user's response to a target. To validate our approach, we conducted an offline validation using previously collected data from a 2D reaching task performed with a nonlinear BoMI. This study represents a first step toward the long-term goal of designing an adaptive BoMI. The adaptation would take place in two asynchronous steps: first the AE decoder is trained to model user's IM, then the AE encoder is trained so as to be the inverse of the user's IM (or, equivalently, the inverse of the AE decoder). This operation would be performed during the online operation of the interface, resulting in a closed-loop adaptation that could potentially improve the ease-of-use of the BoMI itself.

6.2 Methods

6.2.1 AE decoder adaptation

A nonlinear AE is a composition of two sub-networks, an encoder $\boldsymbol{E}$, and a decoder $\boldsymbol{D}$. The procedure for estimating user's IM is referred to as *AE decoder adaptation*, and is described in Fig. 6.1 and Table 6.1.

The ideal model of user's IM should accurately predict the body posture held by BoMI participants in response to the presented target. To approximate such behavior, we consider the 8D vector of IMU signals held by participants a certain time after a single reaching movement started (see section y for further details on this choice). Specifically, the model considers the body signals recorded over m consecutive trials, $\boldsymbol{q}$. The AE decoder is then trained to predict ($\hat{\boldsymbol{q}}$) those body configurations. The input of the decoder is the vector $\boldsymbol{u}$ containing the x and y position of the targets presented over those m trials. Thus, the decoder learns to map the 2D target positions ($\boldsymbol{u}$) over m trials into the m body configurations that the user actually generated. This is achieved by minimizing the mean square error between $\boldsymbol{q}$ and $\hat{\boldsymbol{q}}$:

$$\boldsymbol{J}(\boldsymbol{q},\hat{\boldsymbol{q}}) = \frac{1}{m}\sum_{i=1}^{m}(\boldsymbol{q}_i - \hat{\boldsymbol{q}}_i(\boldsymbol{w},\boldsymbol{b}))^2 \tag{6.1}$$

$\boldsymbol{w}$ and $\boldsymbol{b}$ are the parameters of the AE decoder, representing the matrices of weights and biases of each layer of the network, respectively. The update rule for the decoder parameters is:

$$\boldsymbol{w} \leftarrow \boldsymbol{w} - \alpha\nabla_w\boldsymbol{J}, \qquad \boldsymbol{b} \leftarrow \boldsymbol{b} - \alpha\nabla_b\boldsymbol{J} \tag{6.2}$$

Batch Gradient Descent (BGD) (Ruder, 2016) is used to minimize $\boldsymbol{J}$. Since we assume that users' IM evolves over BoMI training time, we update the decoder paratemers over time as well. Hence, the AE decoder adaptation is designed as an iterative process, in which the parameters are updated via minimization of Eq. 6.1 every time a new target is succesfully completed by the BoMI participant. To do so, at each update iteration (n), $\boldsymbol{q}$ is modified so as to append the vector of body configuration held by the participant during the latest target, and discard that referred to the oldest successful target. Similarly, $\boldsymbol{u}$ is modified so as to append the position of the latest target, and discard that of the oldest one. As a result, both terms act as a moving window. To further facilitate the decoder tracking of user's evolving IM and concurrently encourage smooth updates between consecutive iterations, the decoder

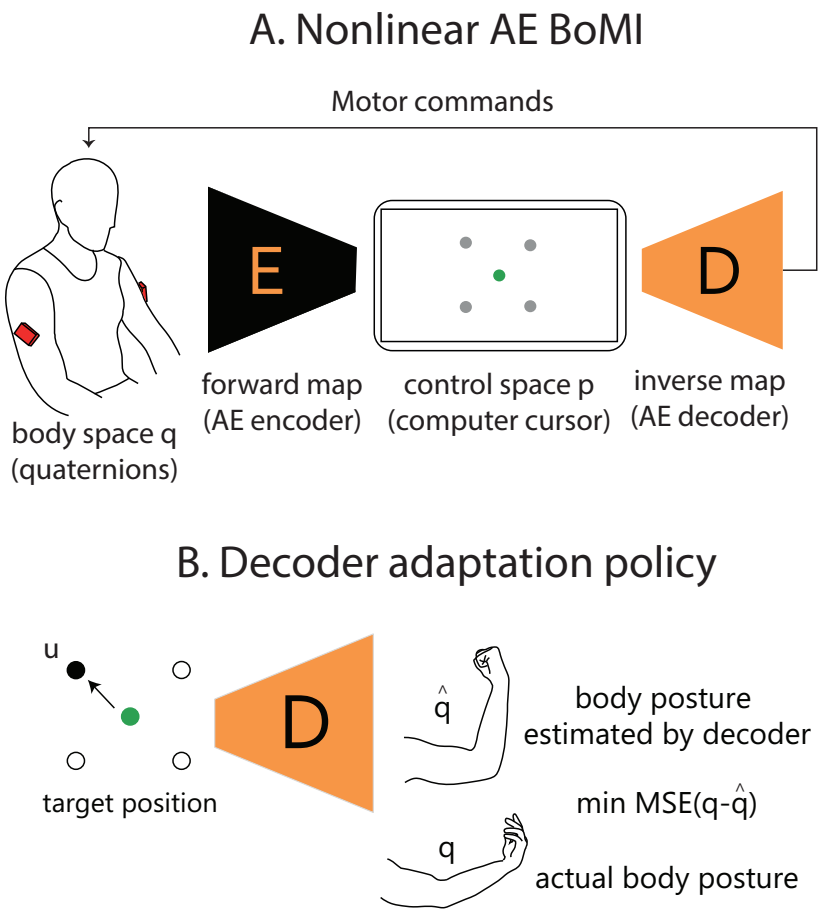

Figure 6.1 AE decoder adaptation policy.

parameters are updated via transfer learning at each iteration (Bengio, 2012). The decoder parameters are initialized following the Xavier normal initialization (Kumar, 2017).

6.2.2 Validation of the model

To validate the proposed AE decoder adaptation approach, we verified whether it was able to perform an offline tracking of the IM of a user that learned to operate a computer cursor via nonlinear BoMI. To do so, we used the data derived from a reaching task whose set-up and protocol were described in Chapter 4 of this thesis. In a nutshell, an AE was used to map the 8D body movements (quaternions from two IMUs placed bilaterally on participants' arms) into cartesian coordinates of a 2D computer cursor. The BoMI forward map consisted of the AE encoder sub-network obtained after a calibration procedure (refer to Section 4.3.4 for

Table 6.1 Pseudo-code for the AE decoder adaptation.

Input: α, $D^{(0)}$, $\boldsymbol{q}^{(0)}$, $\boldsymbol{u}^{(0)}$, $n = 1$
while not terminated **do**:
  if new target i is successful:
    # Step I. update $\boldsymbol{q}$ *and* $\boldsymbol{u}$:
      $\boldsymbol{q}^{(n)} = \text{append}(\boldsymbol{q}^{(n)}, q^{(i)})$, $\text{pop}\left(\boldsymbol{q}^{(n)}[0]\right)$
      $\boldsymbol{u}^{(n)} = \text{append}(\boldsymbol{u}^{(n)}, u^{(i)})$, $\text{pop}\left(\boldsymbol{u}^{(n)}[0]\right)$
    # Step II. Estimate user's IM:
      $\hat{q}^{(n)} = D^{(n-1)}(u^{(n)})$
    # Step III. update decoder parameters by minimizing Eq. 6.1 via BGD
      $w^{(n)} \leftarrow w^{(n)} - \alpha \nabla_w J\left(q^{(n)}, \hat{q}^{(n)}\right)$
      $b^{(n)} \leftarrow b^{(n)} - \alpha \nabla_b J\left(q^{(n)}, \hat{q}^{(n)}\right)$
      $D^{(n)} = f(w^{(n)}, b^{(n)})$
      $n = n+1$, $i = i+1$
  end
end
Output: user's IM estimate at current iteration (n), $\hat{\boldsymbol{q}}^{(n)}$

further details). Ten unimpaired participants performed a reaching task towards a total of 192 targets, placed in four different positions (Fig. 2A).

First and foremost, we had to decide which body posture the AE decoder should predict. In order to be consistent with a recent study by our own group (Pierella et al., 2019), we decided to target the body postures held by participants 1s after the start of a reaching movement. Here, we also expanded the analysis by running the entire procedure separately for postures taken 500 ms and 1.5s after reaching start as well. Thus, a total of three models were trained.

Then, we needed to select the number of reaching movements m to consider during each update iteration (n). While on the one hand a small value of m (*e.g.*, four reaching movements) would allow the model to rapidly track the user's IM, on the other hand it would make the model more prone to overfitting, since it would not contain a sufficiently comprehensive description of user's movement statistic. As a result, the decoder parameters would change too abruptly, a condition that is not ideal especially for a future implementation of this approach online. Thus, one would consider a higher value of m (*e.g.*, twelve reaching movements). In this case, the problem of overfitting would be attenuated, but the model

would be slower at tracking the user's IM. Dealing with this trade-off, we ultimately chose to consider $m = 8$ reaching movement at each update iteration.

Finally, we implemented leave-one-out cross validation by training the model using data related to the first 7 reaching movements and testing it on the data related to the last reaching movement.

Outcome measures

In order to evaluate the quality of the IM developed by the users over training with the BoMI, we defined the **reaching error** (**RE**) as the Euclidean distance between the position of target u and cursor p at a predefined delay. Note that we tested four different models, each with a different target-time after reaching start (500ms, 1s, and 1.5s). Thus, we obtained four RE curves, one specific to each model. (Pierella et al., 2019) showed that training with a linear 2D BoMI leads to a decrease of RE over training time, indicating that BoMI users converged towards a stable (and more accurate) IM. Since the requirement for the online reaching task considered here was to reach the targets as fast as possible, we expected participants to decrease their RE over time. However, they might not have learned to move the cursor closer to the targets within the first 500ms. Thus, we also expected to see different RE trends for each of the three tested target-times. In addition, we fit a single exponential function to the values of RE and computed the coefficient of determination R^2. Higher values of R^2 indicated that participants learned to reduce the RE over online training with the BoMI.

Then, we evaluated the quality of the model of users' IM obtained via AE decoder adaptation. First, for each update iteration (n), we considered the body posture predicted by the decoder ($\hat{q(n)}$) and fed it through the BoMI forward map that was used for the online reaching task. As a result, we obtained the position of the cursor predicted by the model $\hat{p(n)}$. We defined the **inverse model error** (**IME**) as the Euclidean distance between the actual (p) and the predicted ($\hat{p}$) cursor position. Note that this metric considered that users were relying on their IM (as estimated by our model) to reach the targets. Since our model was trained to predict user's IM, and the user's IM was expected to become more stable over training time, we predicted the IME to be lower by the end of the training session. Refer to Fig. 6.2 for a graphical explanation of this condition.

Finally, we computed the ability of the decoder to predict a body posture, starting from the position of the cursor p. For each update iteration (n), we fed $p^{(n)}$ into the current estimate of the IM $(D^{(n)})$, to obtain the vector of body posture $(q_p^{(n)})$ that users would have generated if they were to trust their IM (as estimated by our model) only. The resulting **posture error** (**PE**) was then defined as the Euclidean distance between the actual (q) and

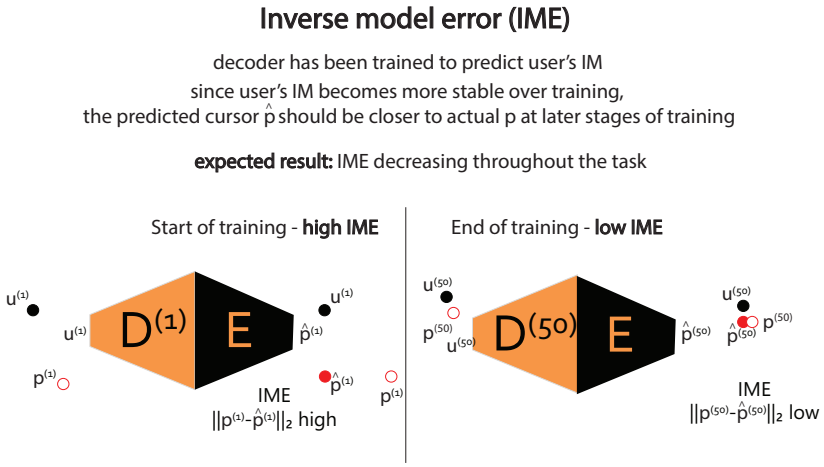

Figure 6.2 Graphical explanation of the inverse model error metric.

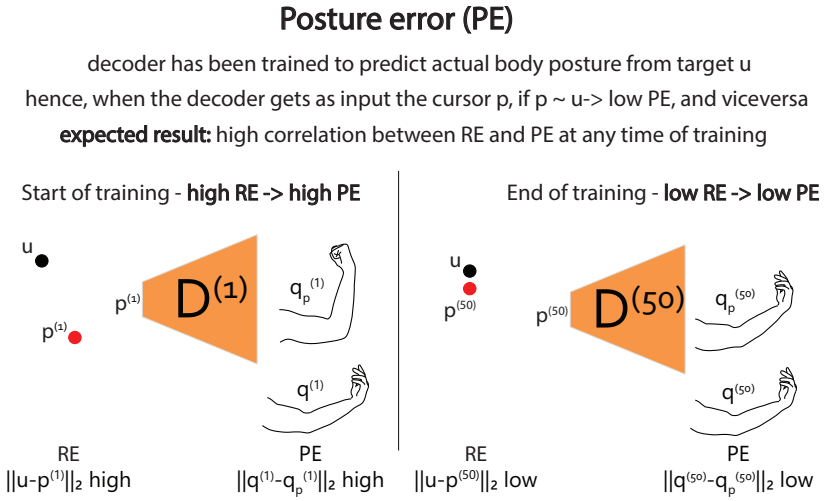

Figure 6.3 Graphical explanation of the posture error metric.

the predicted (q_p) body posture. Note that the decoder was trained to predict body postures from the target position u, not the cursor position p. Thus, the decoder should output a low PE if user's IM is accurate (*i.e.*, low RE, or $p \approx u$). Vice versa, it should output a high PE if user's IM is inaccurate (*i.e.*, high RE, or $p \neq u$). In other words, the ability of the decoder to predict user's IM would correspond to a high correlation between RE and PE at any time of training. Refer to Fig. 6.3 for a visual explanation of this condition.

Note that, at each update iteration, IME and PE were computed using only data from the last successful target (that was not used to train the model due to the leave-one-out crossvalidation).

6.3 Results

Fig. 6.4 shows the reaching error of each participant over online training with the BoMI. During the first 500ms of reaching movements, none of the participants learned to move the cursor closer to the target over training time, as shown by the low values of R^2. However, all participants except one (S1) were able to progressively approach the target faster after 1s and 1.5s since starting reaching movement, with the latter having an overall slightly higher R^2.

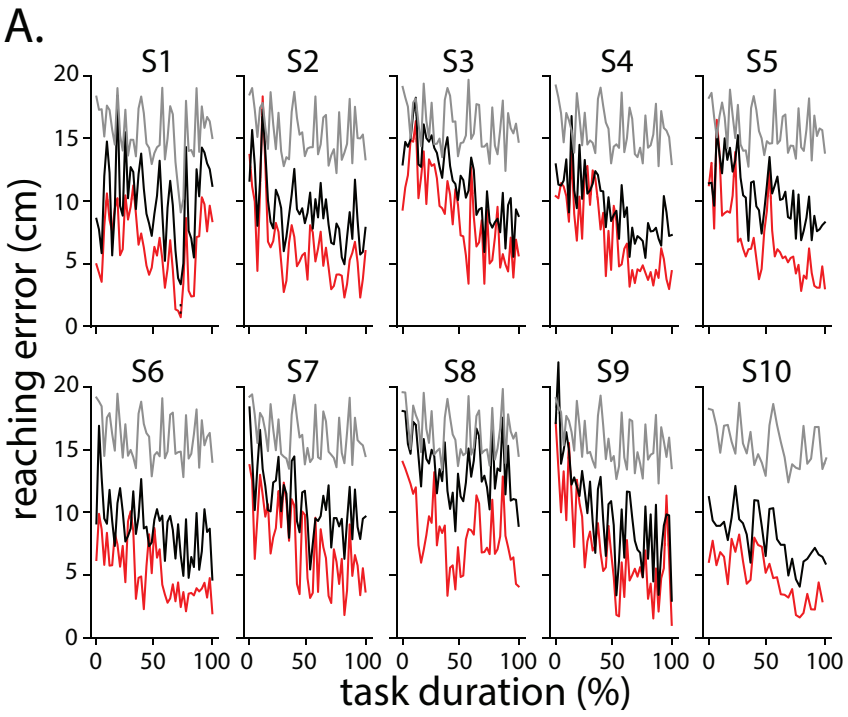

Figure 6.4 Panel A: Reaching error trend over task duration, 500ms (grey), 1s (black) and 1.5s (red) after reaching started. A reduction of RE indicates that participants develoepd a more accurate and stable IM over task duration. Each curve is plotted after applying a moving average filter of 4 reaching movements. Panel B: Table containing the coefficient of determination R^2 after fitting a single exponential on each RE curve.

Fig. 6.5 shows how well the proposed approach succeeds estimating the actual position of the cursor, for each of the three tested target-times. Note that this metric assumed that users were relying on their IM (as estimated by our model) to reach the targets. As shown before, it is clear that our model is not able to accurately predict the cursor position 500ms after reaching start, with no significant accuracy improvements over training time. In contrast, the model of user's IM resulted increasingly accurate over time in predicting the cursor position

both 1s and especially 1.5s after reaching start. In the latter case, excluding participant S1, the prediction error converged around 3cm.

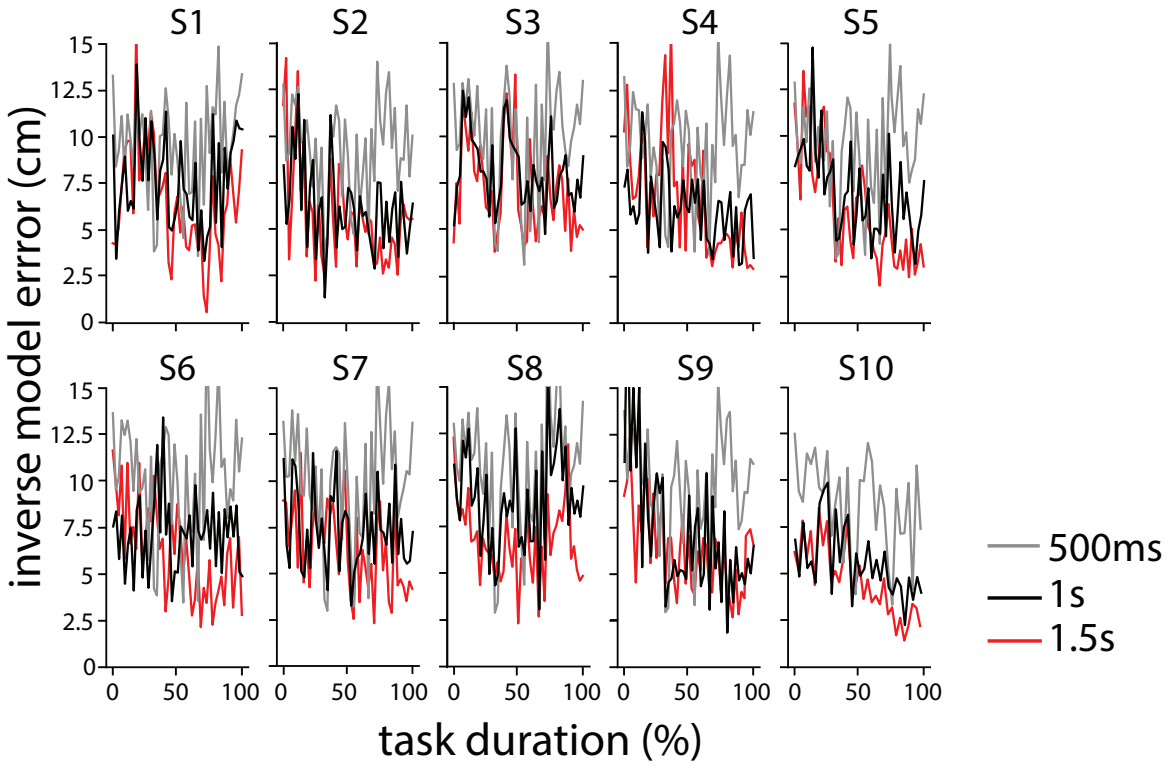

Figure 6.5 Inverse model error 500ms (grey), 1s (black) and 1.5s (red) after reaching started. A reduction of IME indicates that the AE decoder increased its accuracy in predicting the position of the cursor over task duration. Each curve is plotted after applying a moving average filter of 4 reaching movements.

Fig 6.6 shows the posture error and its Pearson correlation coefficient R^2 with the reaching error for each participant. Higher R^2 indicates a greater a accuracy of our model in predicting the actual posture held by the user 500ms, 1s and 1.5s after the reaching movement started. PE and RE did not correlate well by looking at the 500ms target-time, for any of the ten participants. However, increasing the target-time to 1s and 1.5s resulted in higher correlations, indicating that our model was able to capture users' behavior more accurately (with the exception of one participant, S10). In other words, whenever users' IM was not accurate (high values of RE), the corresponding values of PE were high as well, and vice versa. Interestingly, while the AE decoder was not able to predict the cursor position when users' IM did not become more accurate over time (participant S1), it managed to predict their postural behavior instead.

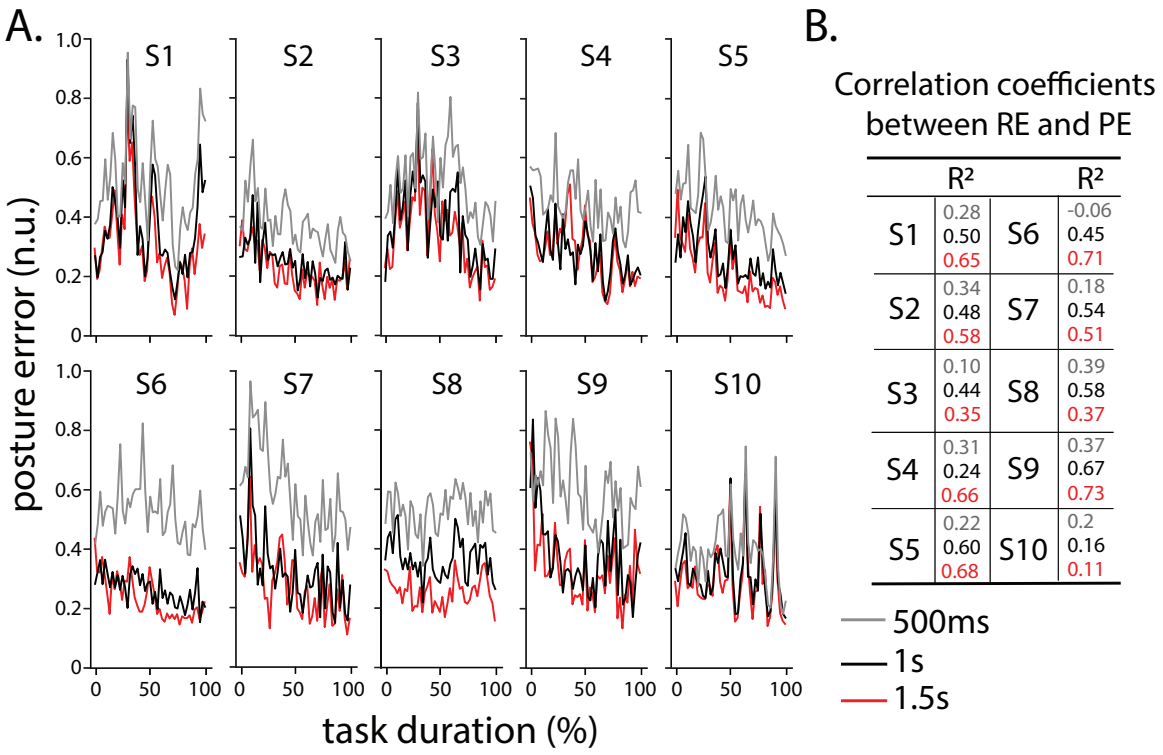

Figure 6.6 Panel A: Posture error 500ms (grey), 1s (black) and 1.5s (red) after reaching started. Each curve is plotted after applying a moving average filter of 4 reaching movements. Panel B: Table containing the Pearson correlation coefficients R^2 between all the curves of reaching error and posture error. Higher R^2 indicates a greater a accuracy of the model in predicting the actual posture held by the user.

Finally, we want to remark that participant S10 completed less reaching movements than all other participants. In the figures above, we resampled each curve so that the first successful reaching corresponded to 0% of task duration and the final successful one to 100% of task duration. Each curve in Fig 6.4, 6.5, and 6.6A was plotted after applying a moving average filter, with a moving window of 4 reaching movements.

6.4 Discussion

This study delivered two main findings: (i) all but one participants developed a more accurate IM over the course of training with a nonlinear BoMI, but not within the first 500ms of movements; (ii) the AE decoder adaptation can accurately track users' behavior, both in terms of posture attained and cursor position one or more seconds after the reaching movement started. The proposed approach seems to be highly dependent on the stability of the user IM, since it produced inaccurate predictions when targeting a time (500ms)

within which participants had not yet developed a stable IM. Nevertheless, we consider these findings supporting a future implementation of our algorithm online for designing an adaptive interface to increase users' performance when operating an external machine. Importantly, our approach to coadaptation is unsupervised, as it does not require external labels and, accordingly, can be performed without interfering with the operation of the interface by the users.

6.4.1 Towards co-adaptation: a two steps approach using AE encoder-decoder

Survivors of spinal cord injury (SCI) or stroke, and those with other disabling conditions, face the combined challenges of regaining functional independence and recovering lost motor skills. The use of human machine interfaces (HMIs) can address both challenges by letting their users controlling an assistive device such as wheelchairs and computers. However, a salient difficulty of current HMIs is the limited ability to match patients' residual motor functions to the devices we try to teach them to control. Current technologies still place the burden of learning to control the device upon their disabled users; the available interfaces do not track and adapt to the evolving skills of their users, which may improve or degrade over time. These barriers limit the efficient and safe use of assistive systems by those most affected in their motor abilities. Thus, the development of new HMI technologies for endowing assistive devices with user-specific adaptive properties is crucial. This goal can be pursued by establishing collaborative reciprocal learning between devices and their users, partially shifting the burden of learning away from the user by adapting the interface to the user's evolving motor abilities.

The idea of adapting a neural interface was first formulated for brain machine interfaces (Carmena et al., 2003; Shenoy and Carmena, 2014; Taylor et al., 2002) based on decoding neural activity to control an external device or computer cursor. It is now widely understood that as the brain learns to reorganize its activity to engage in this novel control task, neural plasticity will trigger changes that render the original interface obsolete (Oweiss and Badreldin, 2015; Shenoy and Carmena, 2014). (Dangi et al., 2013; Orsborn et al., 2014) have proposed a closed-loop adaptation algorithm that updates the interface based on errors with respect to actual or estimated goals. The approach that we want to pursue makes use of similar online methods to design a coadaptive learning body-machine interface that is updated based on the statistics of the motions generated by the user and on the estimate of the user's inverse model.

The basic logic of our coadaptive learning consists in two separate steps and exploits AEs and their encoder-decoder structure. First, as a user develops an inverse model of the BoMI, the AE decoder adaptation procedure proposed in this study allows to adapt the AE decoder becoming more consistent with the user's IM. In this first phase of coadaptation, the AE encoder (*i.e.* the BoMI forward map) is kept fixed according to the parameters found after initial calibration. As the users practice the reaching tasks with the BoMI, a concurrent process trains the AE encoder to approximate an inverse of the function implemented by the decoder. In this second phase of coadaptation, the trained decoder is kept fixed, and subsequent incoming body signal data will be used to retrain and update the encoder. This results in a new encoder that replaces the current BoMI mapping. Through this process, the BoMI will be updated so as to be the inverse of the current estimate of the user's inverse model. Note that this adaptive approach is a gradual online process that runs in parallel with the operation of the interface itself, therefore avoiding disruptions of performance due to BoMI recalibration.

6.4.2 Co-adaptation with AEs cannot be applied at the motor planning stage

A critical question to be resolved is that of timing the interface changes to be most efficient. Updates that are too small or too rare will not facilitate learning, while updates that are too large or too frequent would confuse the user, as an interface perceived as unstable does not facilitate learning (Danziger et al., 2009). This argument suggests the existence of an optimal update schedule that depends upon the users' learning dynamics, more specifically upon the rate at which the users update their own inverse model of the interface. In this sense, a crucial piece of the puzzle is the target time we want to look at when training the decoder to predict the user's response to a target. Ideally, one would want to consider the very first part of a reaching movement, since the immediate result of users movement planning is known to happen within the first 200ms of reaching, when feedback corrections are not yet applied. However, results from this study showed that BoMI users operating a computer cursor via nonlinear AE did not develop a stable IM within the first 500ms of reaching movement (Fig. 6.4). Hence, setting a value that is less than 500ms as the target time for the coadaptation scheduler will most likely prevent the AE decoder to incorporate meaningful information about the state of user's IM (Fig. 6.5).

Therefore, one might want to increase the time horizon for the scheduler to look at, as this has shown to produce more stable predictions over the course of the task duration (also Fig.

6.5). Indeed, this comes at a prize of including more information related to online corrections due to the visual feedback of the cursor when adapting the AE decoder. Although increasing the target time would make the procedure deviates from its original design of incorporating a "purely feedforward" information as the user target response, we argue that a tradeoff in this sense might be necessary.

An alternative approach would be to estimate the user's IM by relying on reaching movements in which the visual feedback of the cursor is turned off. On one hand, this would represent the purest form of incorporating feedforward strategies only. On the other, very few of such blind trials are typically included in a BoMI experimental protocol, as they drastically increase the time constant of learning of the interface. Since the AE decoder adaptation is an empirical procedure, it is highly dependent on the amount of training data used for each update iteration. Hence, reducing the number of data would most likely result in the decoder overfitting a small portion of movement statistic that is not representative of the motor strategies held by the user at each stage of learning.

Chapter 7

Conclusions

In this thesis, I set to expand the knowledge on the implications of using a linear (PCA) vs a nonlinear (AE) dimensionality reduction technique in the context of human-machine interfaces.

First, I confirmed that AEs outperform PCA in their ability to retain more information when asked to reduce the dimension of movement data with minimum loss. Then, I highlighted the reasons underlying such behavior by reinterpreting classic cartographic algorithms in an innovative way. Indeed, I demonstrated that AEs start learning the linear relationship in the input data in the first few epochs of training, effectively matching the performance of PCA. The increased computational power of AEs allows them to further warp their latent manifolds as training goes on so as to capture additional information in the input dataset - a capability that PCA lacks. The proposed cartographic approach also allows to find a consistent latent representation of the input signal, independent of the parameters of the AE model trained to obtain it. I consider these as crucial steps towards the definition of a more rigorous and mathematically accurate approach to promote the use of nonlinear AEs in the context of HMIs.

To further expand on the matter, I directly implemented PCA and AEs as computational engine of a BoMI. I chose to focus on the framework of BoMI since it served the dual purpose of supporting independence of individuals with motor impairment via the operation of assistive devices and of developing therapeutic exercises for the recovery of motor functions.

In particular, I used PCA to create a novel interface that combined muscular and kinematic signals into a hybrid signal for the control of a computer cursor. The addition of the kinematic signal in the control scheme improved the control performance of the interface, fostering its use as therapeutic tool to efficiently target muscle activations, without the limitations associated with EMG-based control alone.

Furthermore, I exploited the increased customizability offered by AEs, with respect to PCA, to design a BoMI for controlling an assistive manipulator with multiple degrees of freedom. In particular, I chose to design a virtual manipulator that that mimicked a commercial robotic arm that can be installed on powered wheelchairs, the MICO arm. All the users practicing with the interface managed after some training to acquire a high level of skill and were able to retain their performance across days. These results serve as a proof of the high potential that AE-based BoMI have as an assistive tool for people with motor impairments.

In the aforementioned studies, the forward map used to derive commands for the external device, remained static over the course of the training after an initial offline tuning. As a result, the users were left with the burden of learning how to use the interface for achieving new goals. Although in both cases users were able to achieve a satisfactory level of skill in operating the external machine, I demonstrated that task-related performance increased with the use of an adaptive interface that updates online to reflect the evolving manifold of user's movements. This indicates that the development of a more accurate internal model is encouraged by online co-adaptation. Importantly, the proposed adaptive interface avoids constraints that other state-of-the-art interfaces commonly have, such as the need of interrupting the operation of the device, relying on information about the state of the task, or depending on the existence of a stable neural or movement manifold.

AEs can be also used as a tool to model user learning. Indeed, I found that it is possible to accurately keep track of the behavior of a user operating a BoMI by gradually updating the AE decoder sub-network so as to predict the user response to a target.

My PhD findings serve as an exploration into the possibility of AE implementation in the context of HMI. The results that emerged over the course of this project highlight the capabilities of AE models as efficient controllers. I hope that this work will act as a guide for other researchers interested in improving assistive technologies for individuals with motor impairments and for those interested in acquiring a stronger understanding of the "black box" mystery of artificial neural networks, such as AEs.

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al. (2016). Tensorflow: A system for large-scale machine learning. In *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*, pages 265–283.

Abdollahi, F., Farshchiansadegh, A., Pierella, C., Seáñez-González, I., Thorp, E., Lee, M.-H., Ranganathan, R., Pedersen, J., Chen, D., Roth, E., et al. (2017). Body-machine interface enables people with cervical spinal cord injury to control devices with available body movements: proof of concept. *Neurorehabilitation and neural repair*, 31(5):487–493.

Adams, M. M. and Hicks, A. L. (2005). Spasticity after spinal cord injury. *Spinal cord*, 43(10):577–586.

Agarap, A. F. (2018). Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*.

Aini, A. and Salehipour, A. (2012). Speeding up the floyd–warshall algorithm for the cycled shortest path problem. *Applied Mathematics Letters*, 25(1):1–5.

Ajoudani, A., Tsagarakis, N., and Bicchi, A. (2012). Tele-impedance: Teleoperation with impedance regulation using a body–machine interface. *The International Journal of Robotics Research*, 31(13):1642–1656.

Anderson, T. W. and Darling, D. A. (1954). A test of goodness of fit. *Journal of the American statistical association*, 49(268):765–769.

Atzori, M., Cognolato, M., and Müller, H. (2016). Deep learning with convolutional neural networks applied to electromyography data: A resource for the classification of movements for prosthetic hands. *Frontiers in neurorobotics*, 10:9.

Baldi, P. and Hornik, K. (1989). Neural networks and principal component analysis: Learning from examples without local minima. *Neural networks*, 2(1):53–58.

Barrese, J. C., Rao, N., Paroo, K., Triebwasser, C., Vargas-Irwin, C., Franquemont, L., and Donoghue, J. P. (2013). Failure mode analysis of silicon-based intracortical microelectrode arrays in non-human primates. *Journal of neural engineering*, 10(6):066014.

Bastos-Filho, T. F., Cheein, F. A., Müller, S. M. T., Celeste, W. C., De La Cruz, C., Cavalieri, D. C., Sarcinelli-Filho, M., Amaral, P. F. S., Perez, E., Soria, C. M., et al. (2013). Towards a new modality-independent interface for a robotic wheelchair. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 22(3):567–584.

Beekhuizen, K. S. (2005). New perspectives on improving upper extremity function after spinal cord injury. *Journal of Neurologic Physical Therapy*, 29(3):157–162.

Belter, J. T., Segil, J. L., Dollar, A. M., and Weir, R. F. (2013). Mechanical design and performance specifications of anthropomorphic prosthetic hands: A review. *Journal of Rehabilitation Research & Development*, 50(5).

Bengio, Y. (2012). Deep learning of representations for unsupervised and transfer learning. In *Proceedings of ICML workshop on unsupervised and transfer learning*, pages 17–36.

Bernstein, N. (1966). The co-ordination and regulation of movements. *The co-ordination and regulation of movements*.

Betke, M., Gips, J., and Fleming, P. (2002). The camera mouse: visual tracking of body features to provide computer access for people with severe disabilities. *IEEE Transactions on neural systems and Rehabilitation Engineering*, 10(1):1–10.

Biddiss, E. A. and Chau, T. T. (2007). Upper limb prosthesis use and abandonment: a survey of the last 25 years. *Prosthetics and orthotics international*, 31(3):236–257.

Bonavita, J., Menarini, M., and Pillastrini, P. (2004). *La riabilitazione nelle mielolesioni*. Elsevier srl.

Bonferroni, C. (1936). Teoria statistica delle classi e calcolo delle probabilita. *Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commericiali di Firenze*, 8:3–62.

Bottou, L. (2010). Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer.

Brucker, B. S. and Buylaeva, N. V. (1996). Biofeedback effect on electromyography responses in patients with spinal cord injury. *Archives of Physical Medicine and Rehabilitation*, 77(2):133–137.

Buongiorno, D., Cascarano, G. D., Camardella, C., De Feudis, I., Frisoli, A., and Bevilacqua, V. (2020). Task-oriented muscle synergy extraction using an autoencoder-based neural model. *Information*, 11(4):219.

Carmena, J. M., Lebedev, M. A., Crist, R. E., O'Doherty, J. E., Santucci, D. M., Dimitrov, D. F., Patil, P. G., Henriquez, C. S., and Nicolelis, M. A. (2003). Learning to control a brain–machine interface for reaching and grasping by primates. *PLoS biol*, 1(2):e42.

Carrington, P., Hurst, A., and Kane, S. K. (2014). Wearables and chairables: inclusive design of mobile input and output techniques for power wheelchair users. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 3103–3112.

Casadio, M., Pressman, A., Fishbach, A., Danziger, Z., Acosta, S., Chen, D., Tseng, H.-Y., and Mussa-Ivaldi, F. A. (2010). Functional reorganization of upper-body movement after spinal cord injury. *Experimental brain research*, 207(3-4):233–247.

Casadio, M., Ranganathan, R., and Mussa-Ivaldi, F. A. (2012). The body-machine interface: a new perspective on an old theme. *Journal of Motor behavior*, 44(6):419–433.

Castellini, C., Artemiadis, P., Wininger, M., Ajoudani, A., Alimusaj, M., Bicchi, A., Caputo, B., Craelius, W., Dosen, S., Englehart, K., et al. (2014). Proceedings of the first workshop on peripheral machine interfaces: Going beyond traditional surface electromyography. *Frontiers in neurorobotics*, 8:22.

Chapin, J. K. and Nicolelis, M. A. (1999). Principal component analysis of neuronal ensemble activity reveals multidimensional somatosensory representations. *Journal of neuroscience methods*, 94(1):121–140.

Chen, Y.-L., Tang, F.-T., Chang, W. H., Wong, M.-K., Shih, Y.-Y., and Kuo, T.-S. (1999). The new design of an infrared-controlled human-computer interface for the disabled. *IEEE Transactions on Rehabilitation Engineering*, 7(4):474–481.

Churchland, M. M., Cunningham, J. P., Kaufman, M. T., Foster, J. D., Nuyujukian, P., Ryu, S. I., and Shenoy, K. V. (2012). Neural population dynamics during reaching. *Nature*, 487(7405):51–56.

Claesen, M. and De Moor, B. (2015). Hyperparameter search in machine learning. *arXiv preprint arXiv:1502.02127*.

Colyer, S. L., Evans, M., Cosker, D. P., and Salo, A. I. (2018). A review of the evolution of vision-based motion analysis and the integration of advanced computer vision methods towards developing a markerless system. *Sports medicine-open*, 4(1):24.

Conditt, M. A., Gandolfo, F., and Mussa-Ivaldi, F. A. (1997). The motor system does not learn the dynamics of the arm by rote memorization of past experience. *Journal of Neurophysiology*, 78(1):554–560.

Cook, A. M. and Polgar, J. M. (2014). *Assistive Technologies-E-Book: Principles and Practice*. Elsevier Health Sciences.

Cowan, R. E., Fregly, B. J., Boninger, M. L., Chan, L., Rodgers, M. M., and Reinkensmeyer, D. J. (2012). Recent trends in assistive technology for mobility. *Journal of neuroengineering and rehabilitation*, 9(1):1–8.

Cremoux, S., Amarantini, D., Tallet, J., Dal Maso, F., and Berton, E. (2016). Increased antagonist muscle activity in cervical sci patients suggests altered reciprocal inhibition during elbow contractions. *Clinical Neurophysiology*, 127(1):629–634.

Cui, P. H. and Visell, Y. (2014). Linear and nonlinear subspace analysis of hand movements during grasping. In *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 2529–2532. IEEE.

Cunningham, J. P. and Byron, M. Y. (2014). Dimensionality reduction for large-scale neural recordings. *Nature neuroscience*, 17(11):1500–1509.

Dangi, S., Gowda, S., Moorman, H. G., Orsborn, A. L., So, K., Shanechi, M., and Carmena, J. M. (2014). Continuous closed-loop decoder adaptation with a recursive maximum likelihood algorithm allows for rapid performance acquisition in brain-machine interfaces. *Neural computation*, 26(9):1811–1839.

Dangi, S., Orsborn, A. L., Moorman, H. G., and Carmena, J. M. (2013). Design and analysis of closed-loop decoder adaptation algorithms for brain-machine interfaces. *Neural computation*, 25(7):1693–1731.

Danziger, Z., Fishbach, A., and Mussa-Ivaldi, F. A. (2009). Learning algorithms for human–machine interfaces. *IEEE Transactions on Biomedical Engineering*, 56(5):1502–1511.

Davidson, P. R. and Wolpert, D. M. (2003). Motor learning and prediction in a variable environment. *Current opinion in neurobiology*, 13(2):232–237.

Dayan, E. and Cohen, L. G. (2011). Neuroplasticity subserving motor skill learning. *Neuron*, 72(3):443–454.

De Santis, D., Dzialecka, P., and Mussa-Ivaldi, F. A. (2018). Unsupervised coadaptation of an assistive interface to facilitate sensorimotor learning of redundant control. In *2018 7th IEEE International Conference on Biomedical Robotics and Biomechatronics (Biorob)*, pages 801–806. IEEE.

De Santis, D. and Mussa-Ivaldi, F. A. (2020). Guiding functional reorganization of motor redundancy using a body-machine interface. *Journal of NeuroEngineering and Rehabilitation*, 17:1–17.

de Silva, V. and Tenenbaum, J. B. (2003). Unsupervised learning of curved manifolds. In *Nonlinear estimation and classification*, pages 453–465. Springer.

Degenhart, A. D., Bishop, W. E., Oby, E. R., Tyler-Kabara, E. C., Chase, S. M., Batista, A. P., and Byron, M. Y. (2020). Stabilization of a brain–computer interface via the alignment of low-dimensional spaces of neural activity. *Nature Biomedical Engineering*, pages 1–14.

Della Santina, C., Piazza, C., Gasparri, G. M., Bonilla, M., Catalano, M. G., Grioli, G., Garabini, M., and Bicchi, A. (2017). The quest for natural machine motion: An open platform to fast-prototyping articulated soft robots. *IEEE Robotics & Automation Magazine*, 24(1):48–56.

Della Santina, C., Piazza, C., Grioli, G., Catalano, M. G., and Bicchi, A. (2018). Toward dexterous manipulation with augmented adaptive synergies: The pisa/iit softhand 2. *IEEE Transactions on Robotics*, 34(5):1141–1156.

DeMers, D. and Cottrell, G. (1992). Non-linear dimensionality reduction. *Advances in neural information processing systems*, 5:580–587.

Di Mattia, P. A., Curran, F. X., and Gips, J. (2001). *An eye control teaching device for students without language expressive capacity: EagleEyes*, volume 53. Edwin Mellen Press.

Dietz, V. (2000). Spastic movement disorder. *Spinal cord*, 38(7):389–393.

DiGiovanna, J., Mahmoudi, B., Fortes, J., Principe, J. C., and Sanchez, J. C. (2008). Coadaptive brain–machine interface via reinforcement learning. *IEEE transactions on biomedical engineering*, 56(1):54–64.

Ditunno, J., Little, J., Tessler, A., and Burns, A. (2004). Spinal shock revisited: a four-phase model. *Spinal cord*, 42(7):383–395.

Donati, A. R., Shokur, S., Morya, E., Campos, D. S., Moioli, R. C., Gitti, C. M., Augusto, P. B., Tripodi, S., Pires, C. G., Pereira, G. A., et al. (2016). Long-term training with a brain-machine interface-based gait protocol induces partial neurological recovery in paraplegic patients. *Scientific reports*, 6:30383.

Donchin, O., Francis, J. T., and Shadmehr, R. (2003). Quantifying generalization from trial-by-trial behavior of adaptive systems that learn with basis functions: theory and experiments in human motor control. *Journal of Neuroscience*, 23(27):9032–9045.

Dosovitskiy, A., Springenberg, J. T., and Brox, T. (2013). Unsupervised feature learning by augmenting single images. *arXiv preprint arXiv:1312.5242*.

Downey, J. E., Schwed, N., Chase, S. M., Schwartz, A. B., and Collinger, J. L. (2018). Intracortical recording stability in human brain–computer interface users. *Journal of neural engineering*, 15(4):046016.

Fan, F., Xiong, J., and Wang, G. (2020). On interpretability of artificial neural networks. *arXiv preprint arXiv:2001.02522*.

Fani, S., Bianchi, M., Jain, S., Pimenta Neto, J. S., Boege, S., Grioli, G., Bicchi, A., and Santello, M. (2016). Assessment of myoelectric controller performance and kinematic behavior of a novel soft synergy-inspired robotic hand for prosthetic applications. *Frontiers in neurorobotics*, 10:11.

Farshchian, A., Gallego, J. A., Cohen, J. P., Bengio, Y., Miller, L. E., and Solla, S. A. (2018). Adversarial domain adaptation for stable brain-machine interfaces. *arXiv preprint arXiv:1810.00045*.

Farshchiansadegh, A., Abdollahi, F., Chen, D., Lee, M.-H., Pedersen, J., Pierella, C., Roth, E. J., Gonzalez, I. S., Thorp, E. B., and Mussa-Ivaldi, F. A. (2014). A body machine interface based on inertial sensors. In *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 6120–6124. IEEE.

Fehr, L., Langbein, W. E., and Skaar, S. B. (2000). Adequacy of power wheelchair control interfaces for persons with severe disabilities: A clinical survey. *Journal of rehabilitation research and development*, 37(3):353–360.

Feldman, A. G. and Latash, M. L. (2005). Testing hypotheses and the advancement of science: recent attempts to falsify the equilibrium point hypothesis. *Experimental Brain Research*, 161(1):91–103.

Fitts, P. M. and Posner, M. I. (1967). Human performance.

Forbes, T. (2013). Mouse hci through combined emg and imu.

Fouad, K. and Tetzlaff, W. (2012). Rehabilitative training and plasticity following spinal cord injury. *Experimental neurology*, 235(1):91–99.

Friedman, M. (1937). The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the american statistical association*, 32(200):675–701.

Fu, Y. and Huang, T. S. (2007). hmouse: Head tracking driven virtual computer mouse. In *2007 IEEE Workshop on Applications of Computer Vision (WACV'07)*, pages 30–30. IEEE.

Gallego, J. A., Perich, M. G., Chowdhury, R. H., Solla, S. A., and Miller, L. E. (2020). Long-term stability of cortical population dynamics underlying consistent behavior. *Nature neuroscience*, 23(2):260–270.

Gallego, J. A., Perich, M. G., Miller, L. E., and Solla, S. A. (2017). Neural manifolds for the control of movement. *Neuron*, 94(5):978–984.

Gallego, J. A., Perich, M. G., Naufel, S. N., Ethier, C., Solla, S. A., and Miller, L. E. (2018). Cortical population activity within a preserved neural manifold underlies multiple motor behaviors. *Nature communications*, 9(1):1–13.

Ganguly, K. and Carmena, J. M. (2009). Emergence of a stable cortical map for neuroprosthetic control. *PLoS Biol*, 7(7):e1000153.

Gao, B. and Pavel, L. (2017). On the properties of the softmax function with application in game theory and reinforcement learning. *arXiv preprint arXiv:1704.00805*.

Gao, P. and Ganguli, S. (2015). On simplicity and complexity in the brave new world of large-scale neuroscience. *Current opinion in neurobiology*, 32:148–155.

Gashler, M., Ventura, D., and Martinez, T. (2008). Iterative non-linear dimensionality reduction with manifold sculpting. In *Advances in Neural Information Processing Systems*, pages 513–520.

Geethanjali, P. (2016). Myoelectric control of prosthetic hands: state-of-the-art review. *Medical Devices (Auckland, NZ)*, 9:247.

Georgi, M., Amma, C., and Schultz, T. (2015). Recognizing hand and finger gestures with imu based motion and emg based muscle activity sensing. In *Biosignals*, pages 99–108.

Gianniotis, N. (2013). Interpretable magnification factors for topographic maps of high dimensional and structured data. In *2013 IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*, pages 238–245. IEEE.

Giggins, O. M., Persson, U. M., and Caulfield, B. (2013). Biofeedback in rehabilitation. *Journal of neuroengineering and rehabilitation*, 10(1):60.

Giuffrida, J. P. and Crago, P. E. (2001). Reciprocal emg control of elbow extension by fes. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 9(4):338–345.

Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256.

Golub, M. D., Sadtler, P. T., Oby, E. R., Quick, K. M., Ryu, S. I., Tyler-Kabara, E. C., Batista, A. P., Chase, S. M., and Byron, M. Y. (2018). Learning by neural reassociation. *Nature neuroscience*, 21(4):607–616.

Grossi, G., Lanzarotti, R., Napoletano, P., Noceti, N., and Odone, F. (2020). Positive technology for elderly well-being: A review. *Pattern Recognition Letters*, 137:61–70.

Gürel, T. and Mehring, C. (2012). Unsupervised adaptation of brain machine interface decoders. *Frontiers in neuroscience*, 6:164.

Haesloop, D. and Holt, B. R. (1990). A neural network structure for system identification. In *1990 American Control Conference*, pages 2460–2465. IEEE.

Hahne, J. M., Biessmann, F., Jiang, N., Rehbaum, H., Farina, D., Meinecke, F. C., Müller, K.-R., and Parra, L. C. (2014). Linear and nonlinear regression techniques for simultaneous and proportional myoelectric control. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 22(2):269–279.

Harburn, K. L. and Spaulding, S. J. (1986). Muscle activity in the spinal cord–injured during wheelchair ambulation. *American Journal of Occupational Therapy*, 40(9):629–636.

Härdle, W. and Vieu, P. (1992). Kernel regression smoothing of time series. *Journal of Time Series Analysis*, 13(3):209–232.

Hargrove, L., Englehart, K., and Hudgins, B. (2006). The effect of electrode displacements on pattern recognition based myoelectric control. In *2006 International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 2203–2206. IEEE.

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. (2016). beta-vae: Learning basic visual concepts with a constrained variational framework.

Hincapie, J. G. and Kirsch, R. F. (2007). Emg-based control for a c5/c6 spinal cord injury upper extremity neuroprosthesis. In *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 2432–2435. IEEE.

Hinton, G. E. and Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507.

Hornik, K., Stinchcombe, M., White, H., et al. (1989). Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366.

Hsu, W.-N., Zhang, Y., and Glass, J. (2017). Unsupervised domain adaptation for robust speech recognition via variational autoencoder-based data augmentation. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 16–23. IEEE.

Hyvärinen, A. and Oja, E. (2000). Independent component analysis: algorithms and applications. *Neural networks*, 13(4-5):411–430.

Ingram, J. N., Körding, K. P., Howard, I. S., and Wolpert, D. M. (2008). The statistics of natural hand movements. *Experimental brain research*, 188(2):223–236.

Jain, S., Farshchiansadegh, A., Broad, A., Abdollahi, F., Mussa-Ivaldi, F., and Argall, B. (2015). Assistive robotic manipulation through shared autonomy and a body-machine interface. In *2015 IEEE international conference on rehabilitation robotics (ICORR)*, pages 526–531. IEEE.

Javanovic, R. and MacKenzie, I. S. (2010). Markermouse: mouse cursor control using a head-mounted marker. In *International Conference on Computers for Handicapped Persons*, pages 49–56. Springer.

Javaremi, M. N. and Argall, B. D. (2020). Characterization of assistive robot arm teleoperation: A preliminary study to inform shared control. *arXiv preprint arXiv:2008.00109*.

Jeong, H., Kim, J.-S., and Son, W.-H. (2005). An emg-based mouse controller for a tetraplegic. In *2005 IEEE International Conference on Systems, Man and Cybernetics*, volume 2, pages 1229–1234. IEEE.

Jordan, M. I. and Rumelhart, D. E. (1992). Forward models: Supervised learning with a distal teacher. *Cognitive science*, 16(3):307–354.

Kaliki, R. R., Davoodi, R., and Loeb, G. E. (2012). Evaluation of a noninvasive command scheme for upper-limb prostheses in a virtual reality reach and grasp task. *IEEE Transactions on Biomedical Engineering*, 60(3):792–802.

Kalman, B. L. and Kwasny, S. C. (1992). Why tanh: choosing a sigmoidal function. In *[Proceedings 1992] IJCNN International Joint Conference on Neural Networks*, volume 4, pages 578–581. IEEE.

Kandel, E. R., Schwartz, J. H., Jessell, T. M., of Biochemistry, D., Jessell, M. B. T., Siegelbaum, S., and Hudspeth, A. (2000). *Principles of neural science*, volume 4. McGraw-hill New York.

Kane, G., Lopes, G., Sanders, J., Mathis, A., and Mathis, M. (2020). Real-time, low-latency closed-loop feedback using markerless posture tracking. *BioRxiv*.

Kawato, M. (1999). Internal models for motor control and trajectory planning. *Current opinion in neurobiology*, 9(6):718–727.

Keim, D. A. (2002). Information visualization and visual data mining. *IEEE transactions on Visualization and Computer Graphics*, 8(1):1–8.

Kim, S., Park, M., Anumas, S., and Yoo, J. (2010). Head mouse system based on gyro-and opto-sensors. In *2010 3rd International Conference on Biomedical Engineering and Informatics*, volume 4, pages 1503–1506. IEEE.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Klambauer, G., Unterthiner, T., Mayr, A., and Hochreiter, S. (2017). Self-normalizing neural networks. In *Advances in neural information processing systems*, pages 971–980.

Koenig, N. and Howard, A. (2004). Design and use paradigms for gazebo, an open-source multi-robot simulator. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, volume 3, pages 2149–2154. IEEE.

Koike, U., Enriquez, G., Miwa, T., Yap, H. E., Kabasawa, M., and Hashimoto, S. (2016). Development of an intraoral interface for human-ability extension robots. *Journal of Robotics and Mechatronics*, 28(6):819–829.

Kornblith, S., Norouzi, M., Lee, H., and Hinton, G. (2019). Similarity of neural network representations revisited. *arXiv preprint arXiv:1905.00414*.

Krakauer, J. W., Ghilardi, M.-F., and Ghez, C. (1999). Independent learning of internal models for kinematic and dynamic control of reaching. *Nature neuroscience*, 2(11):1026–1031.

Kramer, M. A. (1991). Nonlinear principal component analysis using autoassociative neural networks. *AIChE journal*, 37(2):233–243.

Krasoulis, A., Vijayakumar, S., and Nazarpour, K. (2015). Evaluation of regression methods for the continuous decoding of finger movement from surface emg and accelerometry. In *2015 7th International IEEE/EMBS Conference on Neural Engineering (NER)*, pages 631–634. IEEE.

Kullback, S. and Leibler, R. A. (1951). On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86.

Kumar, S. K. (2017). On weight initialization in deep neural networks. *arXiv preprint arXiv:1704.08863*.

Kundu, A. S., Mazumder, O., Lenka, P. K., and Bhaumik, S. (2018). Hand gesture recognition based omnidirectional wheelchair control using imu and emg sensors. *Journal of Intelligent & Robotic Systems*, 91(3-4):529–541.

Lance, J. W. (1980). The control of muscle tone, reflexes, and movement: Robert wartenbeg lecture. *Neurology*, 30(12):1303–1303.

Lauretti, C., Davalli, A., Sacchetti, R., Guglielmelli, E., and Zollo, L. (2016). Fusion of m-imu and emg signals for the control of trans-humeral prostheses. In *2016 6th IEEE International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, pages 1123–1128. IEEE.

Le Cun, Y. (1986). Learning process in an asymmetric threshold network. In *Disordered systems and biological organization*, pages 233–240. Springer.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *nature*, 521(7553):436–444.

Li, G., Schultz, A. E., and Kuiken, T. A. (2010). Quantifying pattern recognition—based myoelectric control of multifunctional transradial prostheses. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 18(2):185–192.

Liao, Q. and Poggio, T. (2017). Theory ii: Landscape of the empirical risk in deep learning. *arXiv preprint arXiv:1703.09833*.

Light, C. M., Chappell, P. H., and Kyberd, P. J. (2002). Establishing a standardized clinical assessment tool of pathologic and prosthetic hand function: normative data, reliability, and validity. *Archives of physical medicine and rehabilitation*, 83(6):776–783.

Lipton, Z. C., Berkowitz, J., and Elkan, C. (2015). A critical review of recurrent neural networks for sequence learning. *arXiv preprint arXiv:1506.00019*.

Liu, W., Wang, Z., Liu, X., Zeng, N., Liu, Y., and Alsaadi, F. E. (2017). A survey of deep neural network architectures and their applications. *Neurocomputing*, 234:11–26.

Losey, D. P., Srinivasan, K., Mandlekar, A., Garg, A., and Sadigh, D. (2020). Controlling assistive robots with learned latent actions. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 378–384. IEEE.

Maaten, L. v. d. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605.

Mahmoudi, B., DiGiovanna, J., Principe, J. C., and Sanchez, J. C. (2008). Co-adaptive learning in brain-machine interfaces. *Brain inspired cognitive systems. Sao Luis, Brazil*.

Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., and Frey, B. (2015). Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*.

Masci, J., Meier, U., Cireşan, D., and Schmidhuber, J. (2011). Stacked convolutional auto-encoders for hierarchical feature extraction. In *International conference on artificial neural networks*, pages 52–59. Springer.

Mathis, A., Mamidanna, P., Cury, K. M., Abe, T., Murthy, V. N., Mathis, M. W., and Bethge, M. (2018). Deeplabcut: markerless pose estimation of user-defined body parts with deep learning. *Nature neuroscience*, 21(9):1281–1289.

Matrone, G. C., Cipriani, C., Carrozza, M. C., and Magenes, G. (2012). Real-time myoelectric control of a multi-fingered hand prosthesis using principal components analysis. *Journal of neuroengineering and rehabilitation*, 9(1):40.

Matrone, G. C., Cipriani, C., Secco, E. L., Magenes, G., and Carrozza, M. C. (2010). Principal components analysis based control of a multi-dof underactuated prosthetic hand. *Journal of neuroengineering and rehabilitation*, 7(1):1–13.

Matsumoto, N., Okada, M., Sugase-Miyamoto, Y., Yamane, S., and Kawano, K. (2005). Population dynamics of face-responsive neurons in the inferior temporal cortex. *Cerebral Cortex*, 15(8):1103–1112.

Mawase, F., Uehara, S., Bastian, A. J., and Celnik, P. (2017). Motor learning enhances use-dependent plasticity. *Journal of Neuroscience*, 37(10):2673–2685.

Maynard, F., Karunas, R., and Waring 3rd, W. (1990). Epidemiology of spasticity following traumatic spinal cord injury. *Archives of physical medicine and rehabilitation*, 71(8):566.

Maynard, F. M., Bracken, M. B., Creasey, G., Ditunno Jr, J. F., Donovan, W. H., Ducker, T. B., Garber, S. L., Marino, R. J., Stover, S. L., Tator, C. H., et al. (1997). International standards for neurological and functional classification of spinal cord injury. *Spinal cord*, 35(5):266–274.

Miehlbradt, J., Cherpillod, A., Mintchev, S., Coscia, M., Artoni, F., Floreano, D., and Micera, S. (2018). Data-driven body–machine interface for the accurate control of drones. *Proceedings of the National Academy of Sciences*, 115(31):7913–7918.

Mintz, J., De Santis, D., Rizzoglio, F., Farshchiansadegh, A., and Mussa-Ivaldi, F. A. (2018). Designing visual feedback to reshape muscle coordination. In *International Conference on NeuroRehabilitation*, pages 1034–1038. Springer.

Morcos, A., Raghu, M., and Bengio, S. (2018). Insights on representational similarity in neural networks with canonical correlation. In *Advances in Neural Information Processing Systems*, pages 5727–5736.

Moro, M., Marchesi, G., Odone, F., and Casadio, M. (2020). Markerless gait analysis in stroke survivors based on computer vision and deep learning: a pilot study. In *Proceedings of the 35th Annual ACM Symposium on Applied Computing*, pages 2097–2104.

Mosier, K. M., Scheidt, R. A., Acosta, S., and Mussa-Ivaldi, F. A. (2005). Remapping hand movements in a novel geometrical environment. *Journal of neurophysiology*, 94(6):4362–4372.

Muceli, S. and Farina, D. (2011). Simultaneous and proportional estimation of hand kinematics from emg during mirrored movements at multiple degrees-of-freedom. *IEEE transactions on neural systems and rehabilitation engineering*, 20(3):371–378.

Muceli, S., Jiang, N., and Farina, D. (2013). Extracting signals robust to electrode number and shift for online simultaneous and proportional myoelectric control by factorization algorithms. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 22(3):623–633.

Mulas, M., Folgheraiter, M., and Gini, G. (2005). An emg-controlled exoskeleton for hand rehabilitation. In *9th International Conference on Rehabilitation Robotics, 2005. ICORR 2005.*, pages 371–374. IEEE.

Müller, J. S., Vidaurre, C., Schreuder, M., Meinecke, F. C., Von Bünau, P., and Müller, K.-R. (2017). A mathematical model for the two-learners problem. *Journal of neural engineering*, 14(3):036005.

Mussa-Ivaldi, F. A., Casadio, M., Danziger, Z. C., Mosier, K. M., and Scheidt, R. A. (2011). Sensory motor remapping of space in human–machine interfaces. In *Progress in brain research*, volume 191, pages 45–64. Elsevier.

Ngeo, J. G., Tamei, T., and Shibata, T. (2014). Continuous and simultaneous estimation of finger kinematics using inputs from an emg-to-muscle activation model. *Journal of neuroengineering and rehabilitation*, 11(1):1–14.

Oby, E. R., Golub, M. D., Hennig, J. A., Degenhart, A. D., Tyler-Kabara, E. C., Byron, M. Y., Chase, S. M., and Batista, A. P. (2019). New neural activity patterns emerge with long-term learning. *Proceedings of the National Academy of Sciences*, 116(30):15210–15215.

Okuma, Y., Mizuno, Y., and Lee, R. G. (2002). Reciprocal ia inhibition in patients with asymmetric spinal spasticity. *Clinical neurophysiology*, 113(2):292–297.

Orban de Xivry, J.-J. and Lefèvre, P. (2015). Formation of model-free motor memories during motor adaptation depends on perturbation schedule. *Journal of Neurophysiology*, 113(7):2733–2741.

Organization, W. H. (2011). World report on disability – 2011.

Orsborn, A. L., Dangi, S., Moorman, H. G., and Carmena, J. M. (2011). Exploring time-scales of closed-loop decoder adaptation in brain-machine interfaces. In *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 5436–5439. IEEE.

Orsborn, A. L., Dangi, S., Moorman, H. G., and Carmena, J. M. (2012). Closed-loop decoder adaptation on intermediate time-scales facilitates rapid bmi performance improvements independent of decoder initialization conditions. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 20(4):468–477.

Orsborn, A. L., Moorman, H. G., Overduin, S. A., Shanechi, M. M., Dimitrov, D. F., and Carmena, J. M. (2014). Closed-loop decoder adaptation shapes neural plasticity for skillful neuroprosthetic control. *Neuron*, 82(6):1380–1393.

Oweiss, K. G. and Badreldin, I. S. (2015). Neuroplasticity subserving the operation of brain–machine interfaces. *Neurobiology of disease*, 83:161–171.

Oxford, R. M. and Daniel, L. G. (2001). Basic cross-validation: Using the" holdout" method to assess the generalizability of results. *Research in the Schools*, 8(1):83–89.

Pandarinath, C., Ames, K. C., Russo, A. A., Farshchian, A., Miller, L. E., Dyer, E. L., and Kao, J. C. (2018a). Latent factors and dynamics in motor cortex and their application to brain–machine interfaces. *Journal of Neuroscience*, 38(44):9390–9401.

Pandarinath, C., O'Shea, D. J., Collins, J., Jozefowicz, R., Stavisky, S. D., Kao, J. C., Trautmann, E. M., Kaufman, M. T., Ryu, S. I., Hochberg, L. R., et al. (2018b). Inferring single-trial neural population dynamics using sequential auto-encoders. *Nature methods*, 15(10):805–815.

Patel, V., Burns, M., Mao, Z.-H., Crone, N. E., and Vinjamuri, R. (2015). Linear and nonlinear kinematic synergies in the grasping hand. *Journal of Bioengineering & Biomedical Sciences*, 5(3):1.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830.

Pierella, C., Abdollahi, F., Farshchiansadegh, A., Pedersen, J., Thorp, E. B., Mussa-Ivaldi, F. A., and Casadio, M. (2015). Remapping residual coordination for controlling assistive devices and recovering motor functions. *Neuropsychologia*, 79:364–376.

Pierella, C., Abdollahi, F., Thorp, E., Farshchiansadegh, A., Pedersen, J., Seáñez-González, I., Mussa-Ivaldi, F. A., and Casadio, M. (2017a). Learning new movements after paralysis: Results from a home-based study. *Scientific reports*, 7(1):1–11.

Pierella, C., Casadio, M., Mussa-Ivaldi, F. A., and Solla, S. A. (2019). The dynamics of motor learning through the formation of internal models. *PLoS computational biology*, 15(12):e1007118.

Pierella, C., De Luca, A., Tasso, E., Cervetto, F., Gamba, S., Losio, L., Quinland, E., Venegoni, A., Mandraccia, S., Muller, I., et al. (2017b). Changes in neuromuscular activity during motor training with a body-machine interface after spinal cord injury. In *2017 International Conference on Rehabilitation Robotics (ICORR)*, pages 1100–1105. IEEE.

Pierella, C., Sciacchitano, A., Farshchiansadegh, A., Casadio, M., and Mussa-Ivaldi, S. (2018). Linear vs non-linear mapping in a body machine interface based on electromyographic signals. In *2018 7th IEEE International Conference on Biomedical Robotics and Biomechatronics (Biorob)*, pages 162–166. IEEE.

Plaut, E. (2018). From principal subspaces to principal components with linear autoencoders. *arXiv preprint arXiv:1804.10253*.

Portnova-Fahreeva, A. A., Rizzoglio, F., Nisky, I., Casadio, M., Mussa-Ivaldi, F. A., and Rombokas, E. (2020). Linear and non-linear dimensionality-reduction techniques on full hand kinematics. *Frontiers in Bioengineering and Biotechnology*, 8:429.

Pressley, A. (2010). Gauss' theorema egregium. In *Elementary differential geometry*, pages 247–268. Springer.

Pylatiuk, C., Schulz, S., and Döderlein, L. (2007). Results of an internet survey of myoelectric prosthetic hand users. *Prosthetics and orthotics international*, 31(4):362–370.

Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., and Ng, A. Y. (2009). Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, page 5. Kobe, Japan.

Raghu, M., Gilmer, J., Yosinski, J., and Sohl-Dickstein, J. (2017). Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. In *Advances in Neural Information Processing Systems*, pages 6076–6085.

Ranganathan, R., Wieser, J., Mosier, K. M., Mussa-Ivaldi, F. A., and Scheidt, R. A. (2014). Learning redundant motor tasks with and without overlapping dimensions: facilitation and interference effects. *Journal of Neuroscience*, 34(24):8289–8299.

Rizzoglio, F., Pierella, C., De Santis, D., Mussa-Ivaldi, F. A., and Casadio, M. (2020). A hybrid body-machine interface integrating signals from muscles and motions. *Journal of Neural Engineering*.

Rizzoglio, F., Sciandra, F., Galofaro, E., Losio, L., Quinland, E., Leoncini, C., Massone, A., Mussa-Ivaldi, F. A., and Casadio, M. (2019). A myoelectric computer interface for reducing abnormal muscle activations after spinal cord injury. In *2019 IEEE 16th International Conference on Rehabilitation Robotics (ICORR)*, pages 1049–1054. IEEE.

Roberts, T. T., Leonard, G. R., and Cepela, D. J. (2017). Classifications in brief: American spinal injury association (asia) impairment scale.

Romero, J., Feix, T., Kjellström, H., and Kragic, D. (2010). Spatio-temporal modeling of grasping actions. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2103–2108. IEEE.

Ruder, S. (2016). An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1985). Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science.

Sadtler, P. T., Quick, K. M., Golub, M. D., Chase, S. M., Ryu, S. I., Tyler-Kabara, E. C., Byron, M. Y., and Batista, A. P. (2014). Neural constraints on learning. *Nature*, 512(7515):423–426.

Sanchez, J. C., Mahmoudi, B., DiGiovanna, J., and Principe, J. C. (2009). Exploiting co-adaptation for the design of symbiotic neuroprosthetic assistants. *Neural Networks*, 22(3):305–315.

Sanger, T. D. (1989). Optimal unsupervised learning in a single-layer linear feedforward neural network. *Neural networks*, 2(6):459–473.

Santello, M., Flanders, M., and Soechting, J. F. (1998). Postural hand synergies for tool use. *Journal of neuroscience*, 18(23):10105–10115.

Santhanam, G., Yu, B. M., Gilja, V., Ryu, S. I., Afshar, A., Sahani, M., and Shenoy, K. V. (2009). Factor-analysis methods for higher-performance neural prostheses. *Journal of neurophysiology*, 102(2):1315–1330.

Scheme, E. and Englehart, K. (2011). Electromyogram pattern recognition for control of powered upper-limb prostheses: state of the art and challenges for clinical use. *Journal of Rehabilitation Research & Development*, 48(6).

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks*, 61:85–117.

Schmit, B. D. and Benz, E. N. (2002). Extensor reflexes in human spinal cord injury: activation by hip proprioceptors. *Experimental brain research*, 145(4):520–527.

Schölkopf, B., Smola, A., and Müller, K.-R. (1997). Kernel principal component analysis. In *International conference on artificial neural networks*, pages 583–588. Springer.

Scholz, J. P. and Schöner, G. (1999). The uncontrolled manifold concept: identifying control variables for a functional task. *Experimental brain research*, 126(3):289–306.

Scholz, M., Fraunholz, M., and Selbig, J. (2008). Nonlinear principal component analysis: neural network models and applications. In *Principal manifolds for data visualization and dimension reduction*, pages 44–67. Springer.

Seáñez-González, I., Pierella, C., Farshchiansadegh, A., Thorp, E. B., Wang, X., Parrish, T., and Mussa-Ivaldi, F. A. (2016). Body-machine interfaces after spinal cord injury: rehabilitation and brain plasticity. *Brain sciences*, 6(4):61.

Segil, J. L. et al. (2013). Design and validation of a morphing myoelectric hand posture controller based on principal component analysis of human grasping. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 22(2):249–257.

Segil, J. L. et al. (2015). Novel postural control algorithm for control of multifunctional myoelectric prosthetic hands. *Journal of rehabilitation research and development*, 52(4):449.

Segil, J. L., Huddle, S. A., et al. (2016). Functional assessment of a myoelectric postural controller and multi-functional prosthetic hand by persons with trans-radial limb loss. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 25(6):618–627.

Shadmehr, R. and Mussa-Ivaldi, F. A. (1994). Adaptive representation of dynamics during learning of a motor task. *Journal of neuroscience*, 14(5):3208–3224.

Shadmehr, R., Smith, M. A., and Krakauer, J. W. (2010). Error correction, sensory prediction, and adaptation in motor control. *Annual review of neuroscience*, 33:89–108.

Shanechi, M. M. (2016). Brain–machine interface control algorithms. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 25(10):1725–1734.

Sharif Razavian, R., Ghannadi, B., and McPhee, J. (2019). On the relationship between muscle synergies and redundant degrees of freedom in musculoskeletal systems. *Frontiers in Computational Neuroscience*, 13:23.

Shenoy, K. V. and Carmena, J. M. (2014). Combining decoder design and neural adaptation in brain-machine interfaces. *Neuron*, 84(4):665–680.

Sherrington, C. S. (1910). Flexion-reflex of the limb, crossed extension-reflex, and reflex stepping and standing. *The Journal of physiology*, 40(1-2):28.

Silva, V. D. and Tenenbaum, J. B. (2003). Global versus local methods in nonlinear dimensionality reduction. In *Advances in neural information processing systems*, pages 721–728.

Sköld, C., Levi, R., and Seiger, Å. (1999). Spasticity after traumatic spinal cord injury: nature, severity, and location. *Archives of physical medicine and rehabilitation*, 80(12):1548–1557.

Sola, J. and Sevilla, J. (1997). Importance of input data normalization for the application of neural networks to complex industrial problems. *IEEE Transactions on nuclear science*, 44(3):1464–1468.

Song, R., Tong, K.-y., Hu, X., and Li, L. (2008). Assistive control system using continuous myoelectric signal in robot-aided arm training for patients after stroke. *IEEE transactions on neural systems and rehabilitation engineering*, 16(4):371–379.

Stopfer, M., Jayaraman, V., and Laurent, G. (2003). Intensity versus identity coding in an olfactory system. *Neuron*, 39(6):991–1004.

Struijk, L. N. A., Egsgaard, L. L., Lontis, R., Gaihede, M., and Bentsen, B. (2017). Wireless intraoral tongue control of an assistive robotic arm for individuals with tetraplegia. *Journal of neuroengineering and rehabilitation*, 14(1):110.

Summa, S., Pierella, C., Giannoni, P., Sciacchitano, A., Iacovelli, S., Farshchiansadegh, A., Mussa-Ivaldi, F. A., and Casadio, M. (2015). A body-machine interface for training selective pelvis movements in stroke survivors: a pilot study. In *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 4663–4666. IEEE.

Sutton, R. S. and Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

Taylor, D. M., Tillery, S. I. H., and Schwartz, A. B. (2002). Direct cortical control of 3d neuroprosthetic devices. *Science*, 296(5574):1829–1832.

Tenenbaum, J. B., De Silva, V., and Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323.

Thompson, B. (2005). Canonical correlation analysis. *Encyclopedia of statistics in behavioral science*.

Thorp, E. B., Abdollahi, F., Chen, D., Farshchiansadegh, A., Lee, M.-H., Pedersen, J. P., Pierella, C., Roth, E. J., Gonzáles, I. S., and Mussa-Ivaldi, F. A. (2015). Upper body-based power wheelchair control interface for individuals with tetraplegia. *IEEE transactions on neural systems and rehabilitation engineering*, 24(2):249–260.

Ting, L. H. and Macpherson, J. M. (2005). A limited set of muscle synergies for force control during a postural task. *Journal of neurophysiology*, 93(1):609–613.

Todorov, E. and Ghahramani, Z. (2004). Analysis of the synergies underlying complex hand manipulation. In *The 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, volume 2, pages 4637–4640. IEEE.

Todorov, E. and Jordan, M. I. (2002). Optimal feedback control as a theory of motor coordination. *Nature neuroscience*, 5(11):1226–1235.

Tropea, P., Monaco, V., Coscia, M., Posteraro, F., and Micera, S. (2013). Effects of early and intensive neuro-rehabilitative treatment on muscle synergies in acute post-stroke patients: a pilot study. *Journal of neuroengineering and rehabilitation*, 10(1):103.

van Middendorp, J. J., Goss, B., Urquhart, S., Atresh, S., Williams, R. P., and Schuetz, M. (2011). Diagnosis and prognosis of traumatic spinal cord injury. *Global spine journal*, 1(1):001–007.

Vellido, A., García, D. L., and Nebot, À. (2013). Cartogram visualization for nonlinear manifold learning models. *Data Mining and Knowledge Discovery*, 27(1):22–54.

Vellido, A., Martín-Guerrero, J. D., and Lisboa, P. J. (2012). Making machine learning models interpretable. In *ESANN*, volume 12, pages 163–172. Citeseer.

Vidaurre, C., Sannelli, C., Müller, K.-R., and Blankertz, B. (2011). Machine-learning-based coadaptive calibration for brain-computer interfaces. *Neural computation*, 23(3):791–816.

Vincent, P., Larochelle, H., Bengio, Y., and Manzagol, P.-A. (2008). Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103.

Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P.-A., and Bottou, L. (2010). Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, 11(12).

Vujaklija, I., Shalchyan, V., Kamavuako, E. N., Jiang, N., Marateb, H. R., and Farina, D. (2018). Online mapping of emg signals into kinematics by autoencoding. *Journal of neuroengineering and rehabilitation*, 15(1):1–9.

Wall, M. E., Rechtsteiner, A., and Rocha, L. M. (2003). Singular value decomposition and principal component analysis. In *A practical approach to microarray data analysis*, pages 91–109. Springer.

Watkins, C., Leathley, M., Gregson, J., Moore, A., Smith, T., and Sharma, A. (2002). Prevalence of spasticity post stroke. *Clinical rehabilitation*, 16(5):515–522.

Wei, Y., Bajaj, P., Scheidt, R., and Patton, J. (2005). Visual error augmentation for enhancing motor learning and rehabilitative relearning. In *9th International Conference on Rehabilitation Robotics, 2005. ICORR 2005.*, pages 505–510. IEEE.

Weiss, E. J. and Flanders, M. (2004). Muscular and postural synergies of the human hand. *Journal of neurophysiology*, 92(1):523–535.

Wilcoxon, F. (1992). Individual comparisons by ranking methods. In *Breakthroughs in statistics*, pages 196–202. Springer.

Wold, S., Esbensen, K., and Geladi, P. (1987). Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52.

Wolpert, D. M., Miall, R. C., and Kawato, M. (1998). Internal models in the cerebellum. *Trends in cognitive sciences*, 2(9):338–347.

Wright, Z. A., Rymer, W. Z., and Slutzky, M. W. (2014). Reducing abnormal muscle coactivation after stroke using a myoelectric-computer interface: a pilot study. *Neurorehabilitation and neural repair*, 28(5):443–451.

Wu, J., Sun, L., and Jafari, R. (2016). A wearable system for recognizing american sign language in real-time using imu and surface emg sensors. *IEEE journal of biomedical and health informatics*, 20(5):1281–1290.

Xiong, A., Chen, Y., Zhao, X., Han, J., and Liu, G. (2011). A novel hci based on emg and imu. In *2011 IEEE International Conference on Robotics and Biomimetics*, pages 2653–2657. IEEE.

Xu, Y., Yang, C., Liang, P., Zhao, L., and Li, Z. (2016). Development of a hybrid motion capture method using myo armband with application to teleoperation. In *2016 IEEE International Conference on Mechatronics and Automation*, pages 1179–1184. IEEE.

Yan, Y., Goodman, J. M., Moore, D. D., Solla, S., and Bensmaia, S. J. (2020). Unexpected complexity of everyday manual behaviors. *bioRxiv*, page 694778.

Yanagisawa, N. et al. (1980). Reciprocal reflex connections in motor disorders in man.

Yeo, S.-H., Franklin, D. W., and Wolpert, D. M. (2016). When optimal feedback control is not enough: Feedforward strategies are required for optimal control with active sensing. *PLoS computational biology*, 12(12):e1005190.

Young, A. J., Hargrove, L. J., and Kuiken, T. A. (2011). Improving myoelectric pattern recognition robustness to electrode shift by changing interelectrode distance and electrode configuration. *IEEE Transactions on Biomedical Engineering*, 59(3):645–652.

Yu, B. M., Cunningham, J. P., Santhanam, G., Ryu, S., Shenoy, K. V., and Sahani, M. (2008). Gaussian-process factor analysis for low-dimensional single-trial analysis of neural population activity. *Advances in neural information processing systems*, 21:1881–1888.

Zecca, M., Micera, S., Carrozza, M. C., and Dario, P. (2002). Control of multifunctional prosthetic hands by processing the electromyographic signal. *Critical Reviews™ in Biomedical Engineering*, 30(4-6).

Zhou, H. and Hu, H. (2008). Human motion tracking for rehabilitation—a survey. *Biomedical signal processing and control*, 3(1):1–18.

Zuo, K. J. and Olson, J. L. (2014). The evolution of functional hand replacement: From iron prostheses to hand transplantation. *Plastic Surgery*, 22(1):44–51.

Appendix A

A Video-Based MarkerLess Body Machine Interface: a Pilot Study

A.1 Introduction

Human disability is a global challenge affecting many people around the world (Organization, 2011). Disability can arise due to a birth condition, an accident or ageing. In this scenario it is necessary to implement and investigate technologies that can improve the quality of life. Assistive Technologies (AT) (Cook and Polgar, 2014) and Positive Technologies (PT) (Grossi et al., 2020) emerge as a powerful solution to address human disability. AT and PT are generic terms for all devices and services that enable the independence of individuals with cognitive and/or functional impairment by improving the conditions of their daily living activities and, consequently, their quality of life.

In this paper we investigate the problem of enabling individuals with motor disabilities (such as after Spinal Cord Injury - SCI) to recover their functional independence. We exploit the fact that, even after a severe injury, many individuals retain some movement, especially of their head and shoulders, that can be used to control external devices, such as a computer cursor. Our approach is based on the framework of Body-Machine Interfaces (BoMIs) (Casadio et al., 2012). BoMIs convert high-dimensional body signals (*e.g.* upper body kinematics, muscle activities) into lower-dimensional, latent, commands to operate an external device. As a result, BoMIs allow individuals with motor disabilities to overcome

The content of this chapter has been published as Moro, M., Rizzoglio, F., Odone, F. and Casadio, M., 2020. A Video-Based MarkerLess Body Machine Interface: a Pilot Study. *ICPR 2020 Workshop Proceedings.*

some of their impairments. The use of BoMIs has been tested in situations involving the control of a computer cursor (Rizzoglio et al., 2020), a powered wheelchair (Thorp et al., 2015) and quadcopters (Miehlbradt et al., 2018). Typically, kinematic-based BoMIs rely on the use of sensors such as inertial measurement units (IMUs) (Pierella et al., 2017a) or markers (Zhou and Hu, 2008) to record the body-movements of their users. For the specific task of cursor control, sensor-based techniques (electrooculargraphy (EOG), electromyography (EMG), IMU, gyro- and opto-sensors) are commonly adopted (Chen et al., 1999; Di Mattia et al., 2001; Jeong et al., 2005; Kim et al., 2010). However, such approaches might hinder the assistive capability of the interface, as sensors cannot be worn autonomously by the BoMI user. Moreover, in the case of training with the BoMI across multiple days, sensors need to be placed consistently so as to minimize the need of interface re-calibration.

Recently, techniques for human motion detection, based on computer vision, have seen a recent surge (Colyer et al., 2018). A video-based marker-less BoMI would potentially allow for a more natural user-friendly interaction with an external device, as it is less invasive and cheaper than a sensor-based one. However, applicability of interfaces that rely only on video-information has seen limited efforts. Javanovic et al. (Javanovic and MacKenzie, 2010) have proposed MarkerMouse, a computer mouse controller based on videos acquired from a webcam that detect a big marker placed on the head of the user. Fu et al. (Fu and Huang, 2007) and Betke et al. (Betke et al., 2002) have developed respectively hMouse and Camera Mouse, two video-based marker-less mouse controllers that detect specific body features, enabling people with severe disabilities to comfortably access a computer, without body attachments.

Here, we want to enhance these approaches combining new computer vision and deep learning techniques and the knowledge derived from the body machine interfaces. Specifically, we present a novel video-based marker-less BoMI pipeline to empower individuals with motor disabilities to independently control a computer cursor via shoulders and/or head movements without the needs of any sensors other than the computer webcam. Our procedure is composed of the following steps (see also Figure A.1): (1) automatic acquisition of images of the user from a computer webcam; (2) detection of landmark points (*e.g.* eyes, nose and shoulders) in the image plane; (3) encoding of the extracted signals to a lower dimensional (control) space via application of a dimensionality reduction (DR) algorithm; (4) handling of the graphic for providing BoMI users with visual feedback of the cursor via a computer monitor. We have evaluated our pipeline in terms of landmark points detection accuracy and overall speed, obtaining encouraging preliminary results. To the best of our knowledge, our

method is the first involving recent state-of-the-art pose estimation algorithm based on deep learning techniques.

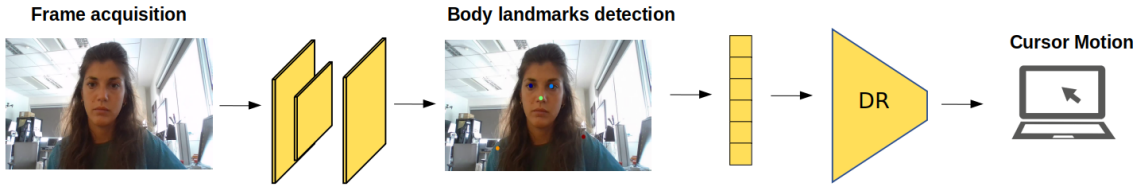

Figure A.1 Summary of the BoMI pipeline. The image acquired by the computer webcam is fed through the trained network to detect the body landmarks. Then, a dimensionality reduction (DR) algorithm is applied to the landmarks' signal to obtain the coordinates of the computer cursor.

A.2 Methods

In this section we present all the building blocks of our pipeline.

A.2.1 Automatic Body Landmarks Detection

The first step of the pipeline is the detection of the positions of body landmarks in the image plane. Since the long-term goal of the pipeline is to empower individuals with motor disabilities, specifically after cervical SCI (cSCI), regaining independence, we decided to focus on the tracking of the body parts whose mobility is most likely retained even after a high level cSCI - *i.e.* shoulders and head (nose and eyes). In this first step, we exploit the DeepLabCut (DLC) architecture (Mathis et al., 2018). DLC is composed by a variant of the Residual Deep Network with 50 layers (ResNet-50) and allows the extraction of specific semantic features after an appropriate fine tuning. The DLC network is pre-trained using the ImageNet dataset. A deconvolutional layer is added at the end to extract spatial density probability maps associated with each landmark point. In order to build a model that correctly detects the position of the body landmarks of interest, we collect a dataset of such landmarks. We ask 40 healthy volunteers to freely move their head and shoulders for 30 seconds so as to comfortably explore their range of motion and to use a computer webcam or a mobile phone to capture a video of such body movements.

Then, we randomly select 15 frames in 32 videos (80% of the total number of videos, for a total of 480 training samples), we manually label the points of interest for each sample and

we fine tune the DLC architecture. The architecture, the optimizer and the hyperparameters of the network are set accordingly to those described in other studies (Mathis et al., 2018; Moro et al., 2020). As a result, the network learn to predict the position (x_i, y_i) of each landmark point in the frame coordinate system, with i= (right eye, left eye, nose, right shoulder, left shoulder). Moreover, DLC returns the likelihood ℓ_i for each $i-$th landmark, expressed as a number in the interval $[0, 1]$ that quantifies the uncertainty behind the detection of each point. We investigate the accuracy of the detection in both train and test videos and we report the results in Section A.3.1.

Figure A.2 shows examples of detection result of this first step. Notice that we include videos with a wide variety of backgrounds, clothes and image dimensions so as to increase the robustness of the model.

Figure A.2 Extracted landmark points (shoulders, nose, eyes) for different subjects.

In order to improve the prediction power of the network and correct for occasional landmarks occlusions, we remove the points with less than 0.7 likelihood. Then, the time sequences of the coordinates for each landmark point are interpolated and low pass filtered (Butterworth, 3rd order, fc=4 Hz).

A.2.2 Encoding Body Landmarks in the 2D Cursor Space

After detecting the body landmarks, the second step consists of applying the BoMI forward map to obtain the (x, y) coordinates of the computer cursor. Since the movements of the nose and the eyes are extremely correlated, we decide to exclude the latter. Thus, the 2D coordinates of shoulders and nose are organized as a 6D vector (q). The BoMI forward map is obtained by asking a volunteer to freely move his head and shoulders for 30s. Then, the DLC model previously trained is applied to the video to extract the vector of body landmarks q for each frame. As a result, a matrix Q containing the estimated coordinates of the landmark points for every frame is obtained. Next, we train a non-linear 2D variational autoencoder (VAE) (Kingma and Welling, 2013) on Q to derive the 2D latent space in which the greatest amount of the body movements variance during calibration is explained. We choose a VAE among the possible methods for dimensionality reduction (DR) (*e.g.*, Principal Component Analysis, *vanilla* AE) due to its ability to enforce a Gaussian distribution within its latent

space. This would ensure a more uniform coverage of the 2D workspace with respect to that obtained training other DR models. To control the uniformity of the latent space, we introduce a scaling term ($\beta = 0.00025$) in the VAE cost function (see (Higgins et al., 2016) for more details). Then, we set the VAE encoder sub-network E as the BoMI forward map. Thus, E maps the 6D body landmark vector (q) into the x-y cursor vector (p):

$$p = E(q) + p_0 \tag{A.1}$$

The offset vector p_0 is chosen to match the origin of the body-space with a corresponding reference position of the cursor. Moreover, the resulting workspace is scaled to ensure full coverage of the computer monitor space (Casadio et al., 2010).

A.2.3 Online Video-Based Marker-Less BoMI

Finally, we set up the online control of a computer cursor with the proposed BoMI. In order to set up the real-time interface, we develop a custom-coded Python script. The script has a multi-threaded architecture so as to handle three different processes: (i) capture the current frame from the computer webcam (via OpenCV library); (ii) forward pass the current frame through the DLC trained model to obtain the body-vector q; (iii) forward pass q through the variational encoder E to obtain the coordinates p to control the cursor (see Figure A.3). During the real-time pipeline, we feed the current webcam frame -read with OpenCV- and the weights of the DLC model to the Deeplabcut-live library (Kane et al., 2020), to obtain a real-time estimation of the body landmarks q. Finally, the encoder E is applied to q in order to obtain the cursor coordinates p for the current frame. To speed up these online operations, we run the code in a computer with a 16 GB NVidia P5000 Quadro GPU.

A.3 Pilot Test and Preliminary Results

In this Section we present the results related to the (1) offline training of the DLC network for the detection of body landmarks in the videos acquired as described in Section A.2.1; (2) real-time operation of the BoMI as described in Section A.2.3.

A.3.1 Body Parts Detection's Accuracy

We evaluate the prediction power of the fine-tuned DLC network by computing the Euclidean distance between the estimates and the manually labeled ground truth positions of the

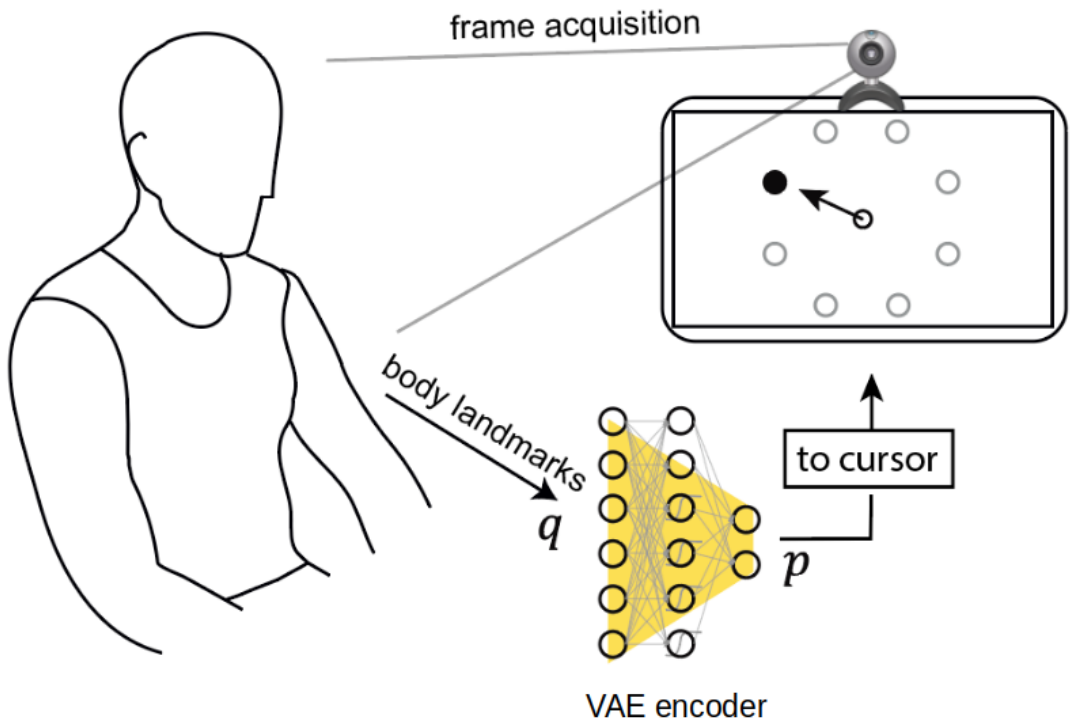

Figure A.3 Scheme of the marker-less BoMI for online cursor control.

landmarks. To do that, we select 5 frames from each of the 32 train videos (160 images) and 5 frames from each of the 8 test videos (40 images). The distance is expressed as pixels. Results show that the network was able to predict the position of the landmarks with a remarkable accuracy (Table A.1).

Table A.1 Mean error ± standard deviation (SD) for each point in pixels computed considering a manually labeled ground truth in 200 images (160 selected from videos adopted to train the network and 40 extracted from test videos).

Point	***Images from Train Videos***	***Images from Test Videos***
Right Eye	2.14 ± 1.12 pxl	2.93 ± 1.36 pxl
Left Eye	2.38 ± 1.08 pxl	2.88 ± 1.25 pxl
Nose	2.29 ± 1.56 pxl	3.04 ± 1.89 pxl
Right Shoulder	3.05 ± 2.03 pxl	3.79 ± 2.88 pxl
Left Shoulder	3.29 ± 2.47 pxl	3.83 ± 2.75 pxl

A.3.2 Online Test of the BoMI

Our goal is to assess whether the BoMI pipeline could be run in real-time. Analyzing the frame recorded by the webcam with the DLC model is a computationally expensive operation, thus achieving a satisfactory frame rate during the online operation of the BoMI is not trivial. We have enrolled a naive unimpaired participants (age 27, male) to practice the online operation of the interface. He has performed a reaching task, in which he is asked to move the cursor over a set of targets as quickly as he can. The position of the cursor and the targets are shown to the participants on a computer monitor. The targets are placed in four different locations, uniformly distributed along a circle. A reaching trial is considered successful after the cursor remains inside the target for 250ms. A total of 192 targets (48 trials per target location) are presented. The order of targets is pseudoranodmized so as each target location is not presented again before all 4 locations have been reached. The participant was immediately able to efficiently move the computer cursor over each target presented. Specifically, he completed all the 192 trials in just 13 minutes. Moreover, we completed all the analysis described in Section A.2.3 with a frame rate of 15 Hz, that allowed a continuous and efficient cursor control.

A.4 Discussion, Conclusion and Future Work

This study delivered three main findings: (i) the fine-tuned DLC network was able to accurately predict the position of body landmarks on images with a variety of different backgrounds and clothes; (ii) such model can be adopted for the online detection of body landmarks; (iii) the proposed pipeline allowed a participant to efficiently and easily operate a computer cursor. For comparison with sensor-based approach, 10 healthy participants practicing cursor control with an IMU-based BoMI completed the same protocol in approximately 20 minutes. Note that this is a pilot study with the aim of exploring the feasibility of the real-time procedure, for this reason only one subject is involved.

The main goal of the study was to verify whether a video-based marker-less BoMI could be used to operate a computer cursor in real-time. The step that takes the most to be completed online is the application of the DLC model to estimate the landmarks position. Only a desktop computer with a powerful GPU would be able to complete this operation within an acceptable time frame during the online cursor control - as we achieved. However, our long-term goal is to improve the pipeline in order to be able to run it on any modern-day laptop, which does not have the same computational capability. This would dramatically

increase the availability of the interface, thus broadening its impact as an assistive device. In order to reach this goal, we would need to reduce the number of the network parameters by adopting other architectures. On one hand, reducing the complexity of the network will allow faster processing, resulting in a more responsive interface. On the other, there is the possibility that it will decrease the accuracy of the estimates, thus leading to a noisier interface. Thus, further testings are required in order to find a correct trade off between computational speed and prediction accuracy. Nevertheless, the interface proposed in this study is already capable of providing a sufficiently intuitive and enjoyable user experience. Therefore, we consider the proposed pipeline adequate for future implementation as an assistive tool for people with motor impairments.

Appendix B

A Myoelectric Computer Interface for Reducing Abnormal Muscle Activations after Spinal Cord Injury

B.1 Introduction

In the immediate aftermath of spinal cord injury, spinal shock causes a hyporeflexia condition that lead to a flaccid paralysis which involves the loss of muscle tone (Ditunno et al., 2004; Sherrington, 1910). Generally, within a few days to a few weeks from the injury, there is a transition to a hyperreflexia condition that might ultimately lead to spasticity (Adams and Hicks, 2005; Dietz, 2000; Lance, 1980). Depending on the level and the severity of the lesion it is possible to predict the likelihood of developing spasticity. For example, for individuals with cervical SCI, the 93% of subjects with a complete injury classified as American Spinal Injury Association (ASIA) 'A' are likely to develop spasticity. This is also the case for 78% of subjects classified as ASIA 'B' to 'D'(Maynard et al., 1990; Sköld et al., 1999). Among the consequences of spasticity, there are changes in the reciprocal Ia inhibition (Okuma et al., 2002; Yanagisawa et al., 1980), that could determine the increase during voluntary contractions of undesired co-activations of antagonist muscles (*i.e.* increase in

The content of this chapter has been published as Rizzoglio, F., Sciandra, F., Galofaro E., Losio L., Quinland E., Leoncini C., Massone A., Mussa-Ivaldi F.A. and Casadio, M., 2020. A Myoelectric Computer Interface for Reducing Abnormal Muscle Activations after Spinal Cord Injury. *2019 IEEE International Conference on Rehabilitation Robotics (ICORR).*

reciprocal facilitation (Cremoux et al., 2016)), and also the activation of muscles belonging to increasingly distant districts, even in the contralateral side (Bonavita et al., 2004). Thus, spasticity affects the ability to perform selective movements and coordinate joints motion, resulting in impaired performance in functional tasks. Moreover, the difficulty of performing a specific movement typically lead SCI individuals to recruit other muscles to compensate for those originally involved in that movement (Beekhuizen, 2005), activating and strengthening undesirable muscle activation patterns. Therefore, it is essential for spinal cord injured individuals to reduce abnormal muscle co-activations within a rehabilitation program aimed at recovery in activities of daily living (ADL).

Myoelectric Computer Interfaces are an effective rehabilitative tool to address these problems. MCIs convert muscle activity, recorded by surface electromyography (sEMG), into control signals operating external devices, including computers. MCIs are frequently used in the prosthetic field, but they have been also used in motor rehabilitation protocols for reducing spasticity in stroke survivors (Song et al., 2008; Wright et al., 2014). One of the factors that have limited the success of MCIs in rehabilitation is their prevalent use for providing mere biofeedback (Giggins et al., 2013). In fact, thinking which muscle to activate and trying to voluntary decrease the activity of the others could increment awareness of individual muscle activity, but it is difficult for people suffering from neurological conditions. We hypothesize that tasks where the goal of decoupling muscle activity is not rendered explicitly, could lead to better results. In this case participants will be motivated and engaged in solving a task aiming at a different goal, implicitly changing their muscle strategies. For example, unimpaired individuals are able to alter their motion strategies (Yeo et al., 2016) or muscle activation patterns (Mintz et al., 2018) pursuing the goal of optimizing the availability of visual information. Therefore, the manipulation of visual feedback could be a key element for MCIs when aiming at implicitly training muscle decoupling. This study aims at providing a first proof of concept for the feasibility of a training protocol based on a MCI aiming at changing muscle activation patterns. Specifically, the long-term goal of the proposed MCI is to reduce undesired muscle co-activations after spinal cord injury by including in the rehabilitative program, in addition to biofeedback of muscle activity, a manipulation of the visual feedback to enforce the desired activity changes.

B.2 Methods

B.2.1 Participants

Six young unimpaired individuals (5 males and 1 female, age 30.50mean±12.63std years) participated in this study. They did not have any evidence or known history of postural, skeletal or neurological disorders and exhibited normal joint range of motion and muscle strength. The interface was also tested on two cervical SCI participants: SCI 1 (level of lesion C5, ASIA B age 44, female), SCI 2 (level of lesion C3, ASIA C age 19, male).

The study conformed to the Declaration of Helsinki and was approved by the local ethical committee (Comitato Etico Regionale - regione Liguria - 201REG2014). All participants signed an informed consent to participate in the study and to the analysis of their data for research purpose.

B.2.2 Experimental Set-up

Surface electromyographic (sEMG) signals were digitally recorded using FREEEMG 1000 system (*BTS Bioengineering Corp., Quincy, MA, USA*). The sampling frequency was 1 KHz. For the unimpaired participants, we recorded the activations of biceps brachii, lateral triceps, and lateral deltoid from the right side of the body. The same muscles were recorded for the SCI 1, but in this case we tested bilaterally. With the other SCI participant (SCI 2) we focused on the lower limb muscles, recording the left rectus femoris as well as the left and right semitendinosus. Each raw signal was preprocessed online to obtain the EMG envelopes used for controlling the position or the visibility of the cursor. Specifically, a bandpass ($f_c = 30 - 450Hz$) 2nd order IIR biquadratic Butterworth filter was used to remove high frequency noise and low frequency motion artifacts. Afterwards, the signal was rectified and a low-pass filter ($f_c = 2Hz$) was applied.

Each session started with a calibration phase where users were asked to perform an isometric Maximum Voluntary Contraction (MVC) task on each muscle for three times. The maximum value obtained in the three repetitions was then saved for further defining the level of activation required in each session. This calibration was followed by a rest phase in which users were asked to fully relax their muscles for 10s. The signals recorded during this resting period were used to subtract from each muscle possible offsets that, while not corresponding to variation on the muscle activations, could have affected the online control of the cursor.

The interface mapped the sEMG from a pair of muscles ($m1$ and $m2$) to two cursor coordinates on the monitor as:

$$x = \frac{sEMG_{m1}}{c_1} * d \tag{B.1}$$

$$y = \frac{sEMG_{m2}}{c_2} * d \tag{B.2}$$

where sEMG indicates the actual value of the EMG envelope, d is the distance (7.2 cm) between the peripheral and the home target in the monitor workspace (see Fig. 1) and c1 and c2 are the scaling values obtained from the MVC for $m1$ and $m2$ respectively at the beginning of each session. Thus $m1$ controlled the horizontal movement of the cursor while $m2$ controlled the vertical movement. A combination of $m1$ and $m2$ moved the cursor through the entire workspace.

For unimpaired participants and SCI 1, we recorded the activity of a third muscle ($m3$) and we controlled the cursor visibility proportionally to its activation. The visibility of the cursor varied from transparent (0%) to maximally visible (100%) as follows:

$$visibility = \begin{cases} 100\%, & m_3 < 5\% \\ 75\%, & 5\% \leq m_3 < 10\% \\ 50\%, & 10\% \leq m_3 < 15\% \\ 25\%, & 15\% \leq m_3 < 20\% \\ 0\%, & m_3 \geq 20\% \end{cases} \tag{B.3}$$

Cursor visibility encoded the additional goal of reducing $m3$ activity while activating $m1$ and $m2$. The interface was developed in C# as a Windows Form Application in Visual Studio 2017, with a targeted framework of .NET 4.5.2. The graphical part of the MCI was developed using OpenTK, a C# graphic library that provided access to graphic tools contained in OpenGL. Graphics refresh rate was set to 50 Hz.

B.2.3 Experimental Protocol

With the proposed MCI, we aimed to address whether individuals with SCI were able to:

- Decouple a pair of muscles ($m1$ and $m2$) that they were co-contracting abnormally;
- Decrease the activity of a muscle ($m3$) that they were concurrently activating with $m1$ and $m2$ due to abnormal strategies.

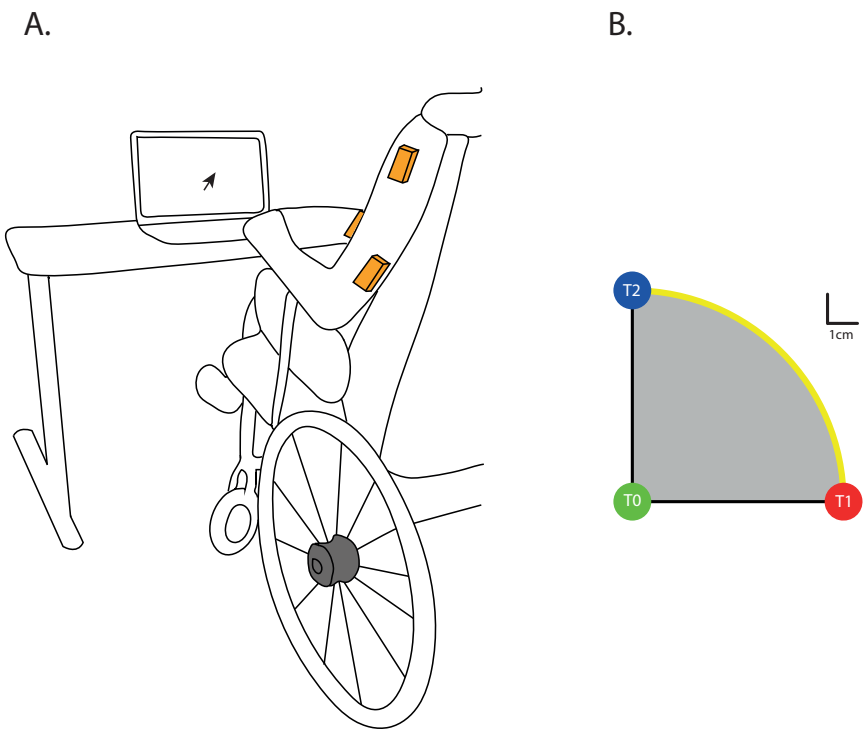

Figure B.1 Panel A: setup for reaching task. Panel B: targets positions on PC monitor. Home Target (green - Tg0) corresponded to the position of the cursor during rest. Target 1 (red - Tg1) was reached by contracting only $m1$. Target 2 (blue - Tg2) was reached by contracting only $m2$.

To validate the interface and identify potential issues that could impact the SCI participants, we tested the MCI on a group of unimpaired participants. The validation was directed at assessing whether unimpaired participants would be able of learning with practice to control the MCI with the independent contraction of two antagonist muscles ($m1$ and $m2$). More importantly, we also verified whether the interface would enforce changes in the activity of a muscle ($m3$) that tended to be naturally coactivated with $m1$ and $m2$.

The protocol was based on the execution of a reaching task. Participants were seated in front of a 15.6" PC monitor at a distance of 1m. The cursor and the targets were shown on the monitor. The participants were asked to reach the targets with straight movements of the cursor. Targets in the workspace were placed as shown in Fig. B.1.

The protocol was divided in three phases:

Training 1

this phase was introduced to let users learn to decouple $m1$ and $m2$. In this phase, the two peripheral targets had to be reached 10 times each in random order. A reaching movement

was considered successful if the cursor remained inside the target for 0.3s. We used an adaptive protocol implicitly asking users to increase the activation of one muscle without a concurrent increase of the other. In fact, the muscular activity needed to reach the target with the cursor gradually increased throughout the training. c1 and c2 cursor scaling values described in equation (1) were initially set at 20% of the corresponding muscle MVC. The subsequent thresholds were respectively set at 25 - 30 - 35 - 40% of the MVC. Note that when the cursor was inside a target the activity of the muscle controlling the opposite direction was close to 0. To increase the threshold for one target, it was necessary to consecutively reach it for three times in less than 60s. Instead, if a target was not reached within 60s, the corresponding threshold was lowered to the previous level.

Training 2

this phase was introduced to let users learn to decrease the activity of $m3$. The protocol was the same as described in Training 1, but here we introduced a variability in the visual feedback given to the users: the more active was the $m3$ muscle, the less visible was the cursor. We hypothesized that users would decrease this muscle activation to maximize the availability of visual feedback.

Test

during this phase users had to reach 18 times each of the two peripheral targets. A reaching movement ended when the cursor remained inside the target for 0.3s or when it remained on any point lying on the arc joining Tg1 and Tg2 (yellow curve shown in Fig. 1) for 1s (being, therefore, at 7.2 cm distance from Tg0). At the end of each movement, we provided users with qualitative feedback of their performance by showing the value of their score (see next section for further details). In this phase, c1 and c2 were set at 40% of the MVC of the corresponding muscle.

B.2.4 Unimpaired participants - Sessions

To validate our approach, we performed a preliminary test on a group of unimpaired participants. Their protocol consisted of 5 phases presented in the order: Test (BASE), Training 1, Test (MID), Training 2, Test (FINAL). They used their right lateral triceps ($m1$) and biceps brachii ($m2$) to control the cursor as well as in Training 2 the right lateral deltoid as $m3$ muscle to modify cursor visibility.

Table B.1 Rehabilitative sessions for SCI 1.

Left (contralateral) muscles		Right (ipsilateral) muscles	
#sess		#sess	
1	Baseline Test	1	Baseline Test
2-3	Test - Training 1 - Test	2-3	Test - Training 1
4-6	Test - Training 2 - Test	4-6	Test - Training 2
7	Final Test	7	Mid Test
		8-11	Test - Training 2
		12	Final Test

Table B.2 Rehabilitative sessions for SCI2.

Left leg muscles		Right-left semitendinosus	
#sess		#sess	
1-3	Test	1-4	Test

B.2.5 SCI participants - Sessions

As for the SCI participants, they were involved in a personalized multisession training, where they were asked to perform isometric contractions of different muscles to move the cursor. The duration of each session was kept short to limit the influence of muscle fatigue on performance.

Specifically, SCI 1 participated in seven rehabilitative sessions for the left arm and twelve for the right arm, with a frequency of three sessions per week. Biceps brachii was used as $m1$, lateral triceps as $m2$ and lateral deltoid as $m3$. Table 1 summarizes the rehabilitative sessions for SCI 1. Each Test session consisted of 10 trials, 5 for each target.

Conversely, the protocol used for SCI 2 had to be strongly customized due to his critical clinical conditions. In particular, he presented severe issues at the respiratory level and high spasticity that sometimes led to the stiffening of the whole body. We, therefore, decided to have him performing only the Test and using it as practice for decreasing co-contraction and increasing awareness of the lower limb muscles. This phase differed from that used for SCI 1 as we increased the number of repetitions for each target from 5 to 18. Three Test sessions were performed for the reduction of left leg muscles co-activation (left semitendinosus as $m1$ and left rectus femoris as $m2$) and four Test sessions for the reduction of pathological co-activation between right ($m1$) and left ($m2$) semitendinosus, with an average frequency of two weekly sessions.

B.2.6 Data Analysis

Three different parameters were chosen to evaluate user's performance:

- **Movement Time** (**MT**): time to reach the peripheral targets after leaving the home target;
- **Smoothness Index** (**NPK**): expressed as the number of peaks (maxima) in the speed profile. We counted the peaks greater than 15% of the maximum speed in each trajectory. This is a measure of smoothness, based on how many cursor sub-movements were included in each reaching movement;
- **Score**: accuracy index reporting on how much the cursor position deviated from the ideal trajectory during a center-out reaching movement. During a trial, every 0.25s we computed the angle (θ) between the straight lines connecting cursor and the target to the starting position. Then, we computed the average of those values at the end of the trial. When $\theta = 90°$ the cursor was totally deviating from the ideal trajectory; thus, in order to have this metric ranging from 0 indicating the worst performance to 100 indicating the best performance, we defined the score as:

$$score = 100 * \left(1 - \frac{\theta}{90}\right) \tag{B.4}$$

  A score of 100 corresponds to the participant activating only $m1(m2)$, while maintaining $m2(m1)$ below its resting value. The score was shown to the user at the end of each trial.

To verify how users were varying the activation of $m3$ we computed:

- **Activation of** $m3$ (**AM3**): % of time over the entire trial duration in which $m3$ was higher than 5% of its MVC.

To test how the indicators related to the motor strategies changed in unimpaired participants we ran repeated measures ANOVA with two within-subjects factors: pratice (1-3: BASE, MID, FINAL for MT, NPK and Score; MID, TRAIN2, FINAL for AM3) and targets (1-2: Tg1, Tg2). A post-hoc analysis (Fisher's LSD test) was used to verify statistically significant differences among factors after repeated measures ANOVA. Significance was asserted when $p < 0.05$. The statistical analysis was computed within Statsoft environment (Statistica software 10.0, Statsoft TULSA, USA).

A single subject design was chosen for SCI participants. Data were analyzed with a visual inspection focusing on the time trend of the measurements and their variability. To verify whether the difference in average between two sets of observations made on the same patient was different from 0, we ran a Wilcoxon rank sum test.

B.3 Results

B.3.1 Unimpaired participants

Repeated measures ANOVA ran on unimpaired participants revealed an effect of training on movement time (MT: $F(2,10) = 7.80$, $p = 0.009$) and smoothness (NPK: $F(2,10) = 11.42$, $p = 0.003$). No significant effects of target positions or interactions with practice were found. Participants became significantly faster and smoother from baseline to the final session (BASE vs FINAL - MT: $p = 0.006$, NPK: $p = 0.002$). They also improved the score (BASE vs FINAL - $p = 0.037$). There was significant learning across the first training block (BASE vs MID - MT: $p = 0.007$; NPK: $p = 0.003$). Additional training with cursor visibility modulation did not lead to further improvement on any performance metric (MID vs FINAL).

As for the activation of the lateral deltoid ($m3$), we found a significant effect of practice (AM3: $F(2,10) = 12.11$, $p = 0.002$). No significant effects of targets position or interactions with practice were found. Specifically, participants significantly decreased lateral deltoid activation during Training 2 with respect to both Mid Test (MID vs TRAIN2 - $p = 0.001$) and Final Test (TRAIN2 vs FINAL - $p = 0.002$). However, no differences were found between activations during Mid and Final Test (MID vs FINAL).

B.3.2 First SCI participant

SCI 1 participated in 7 sessions with the left upper arm. She was able to generally improve the performance from the first to the last session. In particular, the movements became faster (MT: $p_w < 0.001$ for Tg1) and smoother (NPK: $p_w = 0.008$ for Tg2). Concurrently, Score significantly increased when reaching Tg2 ($p_w < 0.001$) but nor for Tg1 ($p_w = 0.236$), while AM3 significantly decreased for both Tg1 ($p_w = 0.001$) and Tg2 ($p_w < 0.001$).

She participated also in 12 sessions with the right upper arm. There was a significant improvement between Baseline and Final Test for MT ($p_w < 0.001$ for both Tg1 and Tg2) and NPK (pw = 0.047 for Tg1, $p_w < 0.001$ for Tg2). Score significantly increased as well, but only for Tg1 ($p_w = 0.002$). She was able to decrease the right lateral deltoid activity at

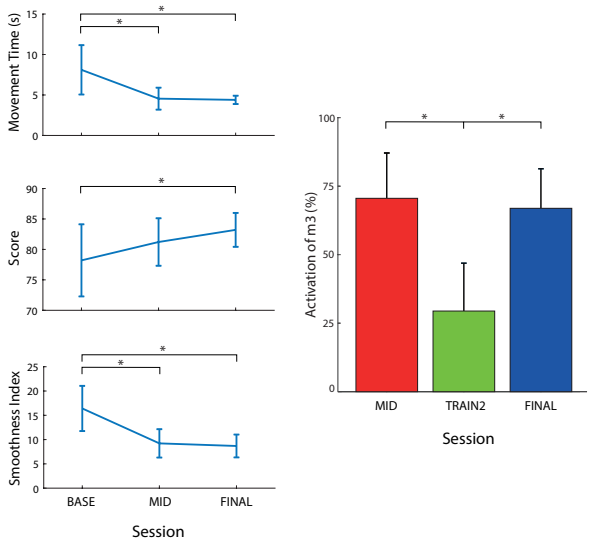

Figure B.2 Panel A: performance metrics values (from top to bottom: MT, Score and NPK) for unimpaired participants during the three sessions of test (BASE, MID and FINAL). Panel B: activation of lateral deltoid ($m3$) during mid test, second session of training (when cursor visibility modulation was active) and final test (red bar indicates MID test, green bar indicates TRAIN2 and blue bar indicates FINAL test). Mean across targets and participants is plotted with standard error.

the end of the rehabilitative session, with a significant difference from the initial session for Tg2 ($p_w < 0.001$), but not for Tg1.

B.3.3 Second SCI participant

SCI 2 participated in three sessions for reducing the co-contraction co-contraction between left rectus femoris and semitendinosus. When reaching Tg2 (*i.e.* when he had to primarily activate the left rectus femoris muscle), he increased the score ($p_w < 0.001$) and moved faster (MT: $p_w = 0.022$) and smoother (NPK: $p_w = 0.001$).

He was also involved in four sessions for reducing the pathological co-activation between left and right semitendinosus. From first to last session, he moved faster (MT: $p_w = 0.05$) and smoother (NPK: $p_w = 0.148$) when moving the cursor towards Tg2, indicating an improved selective activation of the left semitendinosus. Score improved as well ($p_w < 0.001$).

Unfortunately in both cases these positive changes were not observed for Tg1 (*i.e.* respectively when he had to primarily contract the left semitendinosus over the left rectus femoris and the right semitendinosus over the left one).

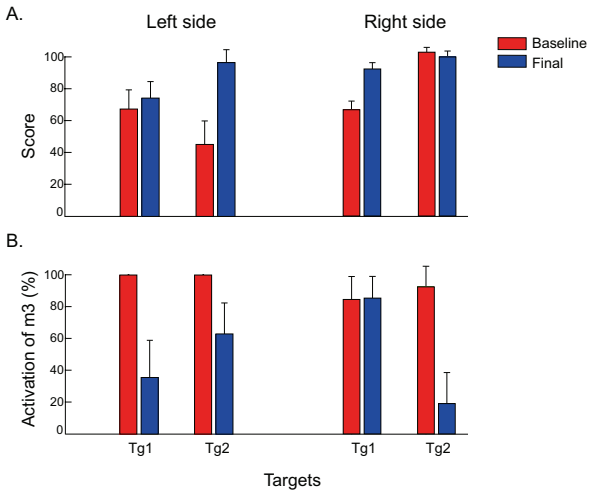

Figure B.3 Panel A: (Left) Score values for left muscles during Baseline (red) and Final (blue) test session and (right) score values for right muscles during Baseline (red) and Final (blue) test session when reaching Tg1 and Tg2. Panel B: (Left) Activation of left lateral deltoid during Baseline (red) and Final (blue) test session and (right) activation of right lateral deltoid during Baseline (red) and Final (blue) test session when reaching Tg1 and Tg2. Mean across repetitions is plotted with standard error.

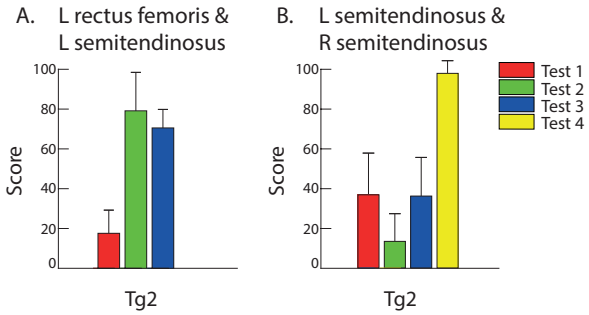

Figure B.4 Panel A: Score values during the three sessions of test for reducing co-activation between left rectus femoris and semitendinosus when reaching Tg2 (first session is plotted in red, second session in green and third session in blue). Panel B: Score values during the four sessions of test for reducing co-activations between left and right semitendinosus when reaching Tg2 (first session is plotted in red, second session in green, third session in blue and fourth session in yellow). Mean across repetitions is plotted with standard error.

B.4 Conclusions

With the proposed MCI we aimed at reducing abnormal muscle simultaneous activations in people with SCI. First, we validated our interface on a group of unimpaired users and then we tested it on two SCI participants by targeting different muscles depending on individual needs.

The results on unimpaired users showed that they improved their performances with practice. Moreover, they decreased the activity of the lateral deltoid but only when cursor visibility was linked with its activation. This confirms the effectiveness of manipulating visual feedback when developing rehabilitation exercises: we think that the retention of this effect may be obtained by increasing the training time. However, this critical conclusion requires additional, more prolonged experiments.

As for the first SCI participant, we tested both the ipsilateral and the contralateral side of the lesion. For the latter, we found a significant improvement from first to last session for all metrics, meaning that she was able to decrease the co-activations between lateral triceps and biceps brachii. For the ipsilateral side, the score did not improve when she had to generate an elbow flexor torque without activating the triceps: this was due to the fact that score was already close to its maximum during baseline. We noticed that the activity of the lateral deltoid by compensatory strategy decreased as well with practice, as she successfully reached a minimum activation during final test, except when she had to primarily contract the ipsilateral triceps.

As for the second SCI participant, we would like to first point out that his was a case of high clinical impairment. We therefore decided to test the interface on the lower limbs, as excessive spasticity affected the upper limb level. We obtained mixed results, as the selective activations of left rectus femoris and semitendinosus improved only when generating an extensor torque of the knee without activating the corresponding antagonist muscle (*i.e.*, semitendinosus). This is consistent with (Schmit and Benz, 2002), as extensor activity generally outlasts flexor activity in SCI due to extensor spasms, that might be triggered by proprioceptors in the hip. Similarly, he reduced the irradiation between left and right semitendinosus only when moving towards the target that required to activate only the left semitendinosus.

Therefore, we concluded that our interface could be used as a valid rehabilitative tool when trying to reduce dysfunctional muscle activations resulting from neurological disorders. We expect that these results will be confirmed in experiments with a larger population of SCI participants.

Acknowledgements

Firstly, I would like to express my sincere gratitude to my advisors Prof. Maura Casadio and Sandro Mussa-Ivaldi for the continuous support and funding of my PhD research. The guidance and immense knowledge they shared helped me throughout the course of my project. Their contagious enthusiasm, dedication, and joy for research was an ongoing motivation for me, and will stick with me as I pursue my future goals.

Besides my advisors, my sincere thanks also goes to all the members (past and present) of the Robotics Lab and the NeuroLab. Three years have past from the first time I participated to an MLB meeting to the completion of this thesis. A big part of my professional improvements goes to you for all the brainstorming sessions and the journal clubs.

Living in Chicago made me a better person, and that is thanks to all the amazing people I have been lucky enough to meet and with whom I shared countless experiences that will remain forever in my memory. To my Italian and International family (Claudio, Valentino, Michel, Linda, Elena, Jeff, Des) - we will celebrate at Kingston Mines again.

During the final year of my PhD, a pandemic has changed our way of living and made many of us unable to visit home. To overcome the apparently insurmountable challenges posed by the pandemic I needed a family. Lorenzo, Maria, Irene, Jacqueline, Sasha, Yaz, and Taras: you will forever be my American family.

To Dalia, who was my first friend in Chicago and my first supporter.

To Courtney, who made me feel home in a foreign land.

To my family. To my Dad and my Mom, who gave me the opportunity to explore the world. I dedicate my work to you.