

NORTHWESTERN UNIVERSITY

Dimensionality Reduction for Prosthetic Hand Control

A DISSERTATION

SUBMITTED TO THE GRADUATE SCHOOL
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

for the degree

DOCTOR OF PHILOSOPHY

Mechanical Engineering

By

Alexandra A. Portnova-Fahreeva

EVANSTON, ILLINOIS

September 2022

© Copyright by Alexandra A. Portnova-Fahreeva

All Rights Reserved

Abstract

In this document, I demonstrate that: 1) Linear basis functions cannot outperform nonlinear ones to represent hand kinematics 2) Nonlinear autoencoders outperform PCA on the dimensionality reduction of hand kinematics, 3) Nonlinear autoencoders outperform PCA in human gait representation and recurrent nonlinear autoencoders can seamlessly express the temporal dynamics, 4) Factors that aid and inhibit one's learning to operate low-dimensional controllers of high-dimensional hand systems, 5) Factors that are important for myoelectric latent representations for low-dimensional control.

Due to the nonlinear nature of hand kinematics during object grasping and gesturing, linear methods, such as Principal Component Analysis (PCA), cannot outperform their nonlinear counterparts as claimed in Yan *et al.* (Yan et al., 2020). Here I present a demonstration of this by applying a simple three-layer nonlinear AE network to Yan's dataset and highlight the superiority of the network over PCA.

I present an analysis of the nonlinear AE network in reducing the dimension of complex hand kinematics and human gait, confirming the superiority of the nonlinear AE structure in its ability to efficiently compress biological data and maintain an equal spread of variance across latent dimensions. I also show that an AE network with a temporal component can perform a more accurate movement classification and individual identification.

Next, I present a low-dimensional myoelectric controller, in which a high-dimensional virtual hand with 17 DOFs is controlled via a 2D space with muscle signals from the wrist. I conduct three studies to understand which factors affect how the user learns to operate such a controller. In particular, short exploration times are insufficient to facilitate learning. Inhibition on the learning

process also happens due to the difficulty of operating a myoelectric interface. Lastly, it is important to provide the users with a clear link between the underlying low-dimensional (2D) controller and the high-dimensional (17D) task of controlling a virtual hand to accelerate learning and achieve the most optimal performances at the end of the training.

After identifying the challenges of learning to operate a low-dimensional controller, I explore what makes a latent space of myoelectric signals useful in the context of low-dimensional controllers. The final study demonstrates that the latent space structure greatly depends on both the input data parameters and the dimensionality-reduction method type.

Acknowledgment

This is to everyone who has stood by my side along this journey – I would not have done it without you.

First of all, I would like to thank my advisor, Sandro Mussa-Ivaldi, who have taken a leap of faith with my enthusiasm for prosthetics and stood by my side along this process regardless of its novelty and deviation from what he was used to for many years of academic research.

A huge thanks to my co- advisor, Eric Rombokas, for, quite literally, adopting me into his research lab and teaching how important it is for a researcher to be boundlessly passionate and over-the-head about their project for one to get to the beautiful places where unexpected discoveries lie. I hope to carry this torch of boundless passion for many years along my side and ignite many other academic souls with the love for research.

I thank my parents and grandparents, who “have known that I will one day become a professor” from a young age and encouraged my interests throughout the school and college years.

A big shoutout to my incredible, knowledgeable labmates, Fabio Rizzoglio and Dalia DeSantis, who have become my secondary advisors. This research would not have been possible without all your help.

Lastly, I would like to thank my husband and my son, who I call “my PhD kid”, for always believing in me, even at the times when I had no belief left for myself. This final stretch would not have been made if not for the two of you.

Table of Contents

Abstract.....	3
Acknowledgment.....	5
Table of Contents	6
List of Tables, Illustrations, Figures, and Graphs	7
Chapter 1 Introduction.....	13
Chapter 2 The Long-Standing Debate of Linear vs Nonlinear Dimensionality-Reduction Methods.....	18
Chapter 3 Linear and Nonlinear Dimensionality-Reduction Techniques on Full Hand Kinematics	23
Introduction	23
Materials and Method.....	26
Results	40
Discussion	51
Chapter 4 Dimensionality Reduction of Human Gait for Prosthetic Control	64
Introduction	64
Materials and Methods	67
Results	75
Discussion	82
Chapter 5 Learning to Operate a High-Dimensional Hand via a Low-Dimensional Controller.....	87
Introduction	87
Methods	90
Results	111
Discussion	121
Chapter 6 Three Challenges of Myoelectric Low-Dimensional Control	129
Introduction	129
Methods.....	131
Results	147
Discussion	155
Chapter 7 Concluding Remarks	162
References	165

List of Tables, Illustrations, Figures, and Graphs

Figure 2-1. Performance of a nonlinear AE (blue) and PCA (red) when applied to complex hand kinematic datasets: American Sign Language (ASL) Gestures, Object Grasps, Activities of Daily Living (ADL) Tasks, and Combined data from the first three datasets (Portnova-Fahreeva et al., 2020).	19
Figure 2-2. Performance of a nonlinear AE network (blue) and PCA (red) on American Sign Language (ASL) gestures and object grasps; data from Yan <i>et al.</i> (Yan et al., 2020).	20
Figure 3-1. 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.....	27
Figure 3-2. Study setup consisting of three different phases: (A) American Sign Language (ASL) Gestures; (B) Object Grasps; (C) Activities of Daily Living (ADL) Tasks.	28
Figure 3-3. The non-linear Autoencoder (AE) network structure used in this study. Curved lines over neurons represent that non-linear activation functions (<i>i.e.</i> , hyperbolic tangent) were used to calculate that layer. Otherwise, activation functions were linear.	32
Figure 3-4. Effects of certain hyperparameters on the non-linear Autoencoder (AE) 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) AE performance difference on training and testing datasets for different activation functions with no regularization. Error bars represent 95% confidence intervals. The AE 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.	34
Figure 3-5. Performance of non-linear Autoencoder (AE, 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. AE outperformed PCA across all datasets and for all number of latent dimensions tested in this study.	41
Figure 3-6. (A) 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 (AE, 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 AE experienced no correlation.	43

- Figure 3-7. Visualization of gesture 1 from American Sign Language Gestures dataset, reconstructed from two-dimensional latent manifolds of non-linear Autoencoder (AE, 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. 44
- Figure 3-8. 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 (AE) and Principal Component Analysis (PCA) and ranked by order of decreasing variance. AE exhibited smaller drop of variance across latent dimensions in comparison to PCA for P1. 45
- Figure 3-9. 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 $p = 0.002$. AE 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. 46
- Figure 3-10. Visualization of the latent trajectories for AE 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). 48
- Figure 3-11. 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 (AE) 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 AE 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 AE with the former being more separable than the latter. 50
- Figure 4-1. Architecture for the Pose-AE network and Move-AE network. Both networks exhibit the classic autoencoder bottleneck shape. Whereas Pose-AE takes single poses as input, Move-AE makes use of stacked recurrent bidirectional LSTM layers to accept entire movement trajectories as input. Both networks embed each input as a single point in a two-dimensional latent space. 71
- Figure 4-2. (A) For each dataset, a randomly chosen sample was reconstructed using PCA and Pose-AE. Pose-AE produced qualitatively improved poses over PCA, though neither was able to reconstruct poses from natural movements dataset, instead electing to reconstruct a mean

standing pose. **(B)** RMSE of joint angles was calculated for each dataset and method. Pose-AE shows a modest improvement over PCA. 76

Figure 4-3. **(A)** For each dataset, a randomly chosen sample was reconstructed using PCA and Pose-AE. Pose-AE produced qualitatively improved poses over PCA, though neither was able to reconstruct poses from natural movements dataset, instead electing to reconstruct a mean standing pose. **(B)** RMSE of joint angles was calculated for each dataset and method. Pose-AE shows a modest improvement over PCA. 77

Figure 4-4. The three panes display the best performing latent space of their respective method. The latent space is a visualization of the activations of the two coding units in the bottleneck, or the first two principal components. The bottom-left pane shows the results of a movement classifier SVM trained on the latent spaces for each method. Error bars denote the varying performance from each of 10 runs for both autoencoders. The Move-AE latent space outperformed Pose-AE and PCA latent spaces on classifying between flat ground and stair walking. Recall that in the Move-AE latent space, each point represents an entire movement. Flat ground walking is embedded in a cyclical structure that is well separated from stair walking.... 79

Figure 4-5. The three panes display the best performing latent space of their respective method. The plotted data is colored by one of eight individual subjects. The classification error for the best performing model is included in the top-right of each plot. The bottom-left pane shows the results of an individual-specific classifier SVM trained on the latent spaces for each method. Note that the Move-AE latent space here is a well-performing outlier, though visual inspection of the other latent spaces confirms the general behavior of separately embedding individuals. 81

Figure 5-1. Experimental setup. **(A)** Study I had a long exploration window of 60s and implicit target-gesture training via a myoelectric interface. **(B)** Study II had a long exploration window of 60s and implicit target-gesture training via a mouse-based interface. **(C)** Study III incorporated explicit target-gesture training with a long exploration window of 60s and a mouse-based interface. 89

Figure 5-2. **(A)** Structure of a regular autoencoder (AE) with three hidden layers. The middle layer represents the latent space. Curves over the neurons represent layers obtained with a nonlinear activation function. **(B)** Structure of a variational autoencoder (VAE) with three hidden layers and a regularizer term before the latent space. **(C)** Latent space derived by applying a variational autoencoder to hand kinematics data of an individual performing American Sign Language (ASL) gestures. 91

Figure 5-3. **(A)** 17 degrees of a freedom (DOFs) of the virtual hand. **(B)** Surface electrode placement on the participant's forearm in the myoelectric interface. Each electrode location is specific to one of the four wrist movements. **(C)** Experiment setup with the participant sitting approximately 1.5m away from the computer screen during one of the studies. **(D)** Eight American Sign Language (ASL) gestures that the participants were being trained to recreate during the studies. 95

Figure 5-4. Setup of a myoelectric interface. Wrist movements generated EMG signals, which, in turn, were combined using a Vector Summation Algorithm into a 2D vector (X_{EMG}, Y_{EMG}). The vector, in turn, controlled the position of a 2D cursor on the latent space. Every point on the

latent space (X_{kinem}, Y_{kinem}) reconstructed into full 17D hand kinematics via the decoder part of the variational autoencoder network. 97

Figure 5-5. (A) Setup of the 17D task with two virtual 17D hands present. The hand on the left is the target hand that the participants needed to match. The hand on the right is the controlled hand that the participants controlled either via a myoelectric or a mouse interface. (B) Setup of the $2D_{implicit}$ task where the participants performed simple 2D reaches – with the blue controlled cursor and the overlaid hand to the red target and its corresponding hand. (C) Setup of the $2D_{explicit}$ task, in which the participants controlled the blue cursor on the 2D plane, which in turn controlled the 17D of the hand on the right. The hand on the left was the target hand. The participants were required to learn which of the eight present red targets represent the 2D location of the target gesture. (D) Sequence of training and test session in each study. The only difference between the groups is in *Train1* session, where the 17D group only experiences the 17D task while the $2D_{implicit/explicit}$ groups have the session split in half: $2D_{implicit/explicit}$ task and 17D task. 101

Figure 5-6. Sampling of modified gestures from the latent space. Modified gestures were sampled from 75% of the nominal path between the neutral position and the complete gesture on the latent space. 107

Figure 5-7. Outcome measures of Study I. (A) Average success rate for each session across groups. (B, left) Average adjusted reach time (ART) for every repetition across all training sessions for each group. (B, right) Average ART for the first 8 repetitions of the 17D task for each group. (C, left) Average adjusted path efficiency (APE) for every repetition across all training sessions for each group. (C, right) Average APE for the first 8 repetitions of the 17D task for each group. Green colors represent different training/test sessions of the $2D_{implicit}$ group. Magenta colors represent different training/test sessions of the 17D group. Error bars represent standard errors. Black asterisks (*) signify statistical differences *between* the groups. Magenta asterisks (*) represent statistical differences *within* the 17D group. Green asterisks (*) represent statistical differences *within* the $2D_{implicit}$ group. 113

Figure 5-8. Outcome measures of Study II. (A) Average success rate for each session across groups. (B, left) Average adjusted reach time (ART) for every repetition across all training sessions for each group. (B, right) Average ART for the first 8 repetitions of the 17D task for each group. (C, left) Average adjusted path efficiency (APE) for every repetition across all training sessions for each group. (C, right) Average adjusted APE for the first 8 repetitions of the 17D task for each group. Green colors represent different training/test sessions of the $2D_{implicit}$ group. Magenta colors represent different training/test sessions of the 17D group. Error bars represent standard errors. Magenta asterisks (*) represent statistical differences *within* the 17D group. Green asterisks (*) represent statistical differences *within* the $2D_{implicit}$ group. 117

Figure 5-9. Outcome measures of Study III compared to Study II ($2D_{implicit}$ group only). (A) Average success rate for each session across groups. (B) Average adjusted reach time (ART) for every repetition across all training sessions for each group. (C) Average adjusted path efficiency (APE) for every repetition across all training sessions for each group. Green lines represent different training/test sessions of the $2D_{implicit}$ group. Blue lines represent different training/test sessions of the $2D_{explicit}$ group. Blue asterisks (*) represent statistical significance within the

2D_{explicit} group. Black double asterisks (**) represent statistical significance between the 2D_{explicit} and 2D_{implicit} groups..... 119

Figure 6-1. (A) Principal component analysis (PCA) achieves dimensionality reduction by finding the directions along which the input data vary the most and projecting the data along them. (B) Autoencoder (AE) network performs dimensionality reduction by compressing the input into the latent space via the encoder and reconstructs the output via the decoder. (C) Variational autoencoder (VAE) network performs dimensionality reduction, similar to AE, and regularizes its encodings distribution during training in order to ensure that its latent space has a target distribution. (D) Latent space of hand kinematics generated by applying a VAE on the data from a participant performing American Sign Language number gestures. Different colors represent various number gestures..... 133

Figure 6-2. Visualization of latent space factors. (A) Class separability defined as a measure of how separable points of different classes on a latent space are with a straight line. (B) Location of the resting muscle signals, EMG_{rest} , encoded on a latent space. EMG_{rest} location is found by first finding the minimum-area ellipse that includes all points on the latent space (blue). The ellipse is then transformed to a unit circle (magenta). The location of EMG_{rest} is calculated by finding the Euclidian norm between the transformed EMG_{rest} and the center of the unit circle. (C) Latent dimension variance is defined how much variance each of the dimension has with respect to the sum of variances of both dimensions. Latent dimension variance spread is defined as the difference between the dimensions of maximum and minimum variances..... 141

Figure 6-3. Visualization of 2D latent spaces produced by principal component analysis (PCA), autoencoder (AE) network, and variational autoencoder (VAE) network. Latent spaces are generated on the unimpaired (first row) and amputee (second row) datasets. The AE- and VAE-generated latent spaces are obtained with a fixed seed parameter. Grasps 1-5 are indicated by light blue, dark blue, purple, and pink colors, respectively. Both datasets were recorded in a *dynamic sit* condition, and the performed grasps were precision disk, prismatic pinch, index finger extension, adducted thumb, prismatic four-finger grasps. Black datapoints represent resting EMG signals encoded onto the latent space (EMG_{rest}). Sep is the class separability factor, v_{spread} is the spread of dimension variance, and Loc_{rest} is the location of the resting EMG, EMG_{rest} 148

Figure 6-4. Stability of latent space factors (class separability, resting EMG, EMG_{rest} , location on the low-dimensional space, and spread of variance of two latent dimensions) for the unimpaired and amputee datasets. (A) Stability of class separability, EMG_{rest} location, dimension variance for unimpaired dataset considering random seed factor for autoencoder (AE) and variational autoencoder (VAE) networks. (B) Stability of class separability, EMG_{rest} location, dimension variance for the amputee dataset considering random seed factor for AE and VAE networks. Blue points indicate resulting factor values for the AE network. Magenta points indicate resulting factor values for the VAE network. Circles outlined in black indicate mean values while the errorbars signify standard deviation..... 150

Figure 6-5. Visualization of the 2D latent spaces produced by principal component analysis (PCA), autoencoder (AE) network, and variational autoencoder (VAE) network. Latent spaces are generated on the low-density (first row) and high-density (second row) datasets. The AE- and VAE-generated latent spaces are obtained with a fixed seed parameter. Flexion, extension,

adduction, and abduction of the wrist are indicated by light blue, dark blue, purple, and pink colors, respectively. Black datapoints represent resting EMG signals encoded onto the latent space (EMG_{rest}). Sep is the class separability factor, v_{spread} is the spread of dimension variance, and Loc_{rest} is the location of the resting EMG, EMG_{rest} 151

Figure 6-6. Stability of latent space factors (class separability, resting EMG, EMG_{rest} , location on the low-dimensional space, and spread of variance of two latent dimensions) for the high- and low-density datasets. **(A)** Stability of class separability, EMG_{rest} location, dimension variance for the high-density dataset considering random seed factor for autoencoder (AE) and variational autoencoder (VAE) networks. **(B)** Stability of class separability, EMG_{rest} location, dimension variance for the low-density dataset considering random seed factor for AE and VAE networks. Blue points indicate resulting factor values for the AE network. Magenta points indicate resulting factor values for the VAE network. Circles outlined in black indicate mean values while the errorbars signify standard deviation..... 154

Figure 6-7. Correlation between muscle signals for different datasets. **(A)** Heatmaps of correlation between 12 acquired EMG signals for the *acquisition conditions dataset*. The four conditions are *dynamic sit*, *dynamic stand*, *static sit*, and *static stand*. **(B)** Heatmap of correlation between 12 acquired EMG signals for the *amputee dataset*. **(C)** Heatmap of correlation of 12 acquired EMG signals for the *low-density dataset*. **(D)** Heatmap of correlation of 256 EMG signals for the *high-density dataset*. **(E)** Average values EMG correlation (EMG_{corr}) for each of the dataset. The errorbars represent 95% confidence intervals. Double asterisk (**) indicates that the datasets yield EMG_{corr} values that are statistically different than any other dataset values. Single asterisk (*) indicates significant difference in EMG_{corr} values between the datasets. 155

Chapter 1 Introduction

It has been nearly 70 years since the development of the first myoelectric prosthetic hand (Reiter, 1948). Since then, the field has seen many advancements: from intricate finger designs (Carrozza et al., 2002; Weghe et al., 2004; Gaiser et al., 2009; Hackett et al., 2013; Zuo and Olson, 2014) to complex control systems that incorporated emerging machine learning (ML) methods (Jiang et al., 2008; Scheme and Englehart, 2011; Malhotra et al., 2012; Rehbaum et al., 2012; Young et al., 2012; Benatti et al., 2017; Hahne et al., 2018). But despite the amount of effort that has been put into the development of usable prosthetic hands, the preferred method of hand replacement remained the same as it has been for many decades – a traditional split hook (Østlie et al., 2012). The paradox of that is staggering, yet clear when one understands, first, the ability of a unilateral amputee (the most common in arm amputations) to perform daily tasks single-handedly, and, second, the complexity of a human hand that a prosthesis must replace.

Current prosthetic solutions, although intricate in their design and availability of many degrees of freedom (DOFs), lack a simple and intuitive control method. When dealing with myoelectric interfaces on amputees, availability of clean electromyographic (EMG) signals that could be usable for a control purpose is scarce. As a result, one is left with just a few signals to control a plethora of prosthetic DOFs – not an easy challenge at hand, actually and figuratively.

The solutions to dealing (or rather avoiding) the mismatch problem between the controlling EMG signals and the controlled DOFs have been many: from finite-state machines (FSM) (Kyberd et al., 1995; Cipriani et al., 2008) to complex pattern recognition (PR) (Scheme and Englehart, 2011; Young et al., 2012; Kuiken et al., 2016). In the former, two EMG signals are used to switch between and select grip patterns. In the latter, up to eight muscle signals are used to classify a set

of unique contractions that are matched to specific preselected grasps. In both cases, the controller can become strenuous and mentally-fatiguing when presented with a large number of hand postures.

An attempt to deal with the dimensionality mismatch in prosthetic control has been performed by several groups, who reduced the dimensionality of complex hand kinematics via a linear dimensionality-reduction algorithm, called principal component analysis (PCA) (Magenes et al., 2008; Ciocarlie and Allen, 2009; Matrone et al., 2010; Matrone et al., 2012; Segil, 2013; Segil and Controzzi, 2014; Segil, 2015; Segil and Huddle, 2016). Simply speaking, PCA finds the directions in which the input data vary the most and projects the data along these directions, thus obtaining a low-dimensional, *latent*, space. In the past, PCA has been applied to hand kinematics during object grasping and proven to be an effective tool of reducing the dimensionality of kinematic signals (Santello et al., 1998; Todorov and Ghahramani, 2004; Ingram et al., 2008). Building on these results, several research groups have utilized PCA to obtain a latent space to perform a control of a high-dimensional system (*e.g.*, virtual or prosthetic hand) via a low-dimensional controller. Unlike the classifiers used in PR, PCA has the advantage of being an *unsupervised* ML algorithm, which does not rely on labeled data and does not require any time-consuming training. In addition, one could argue that the method harnesses one's *natural* way of controlling body parts as it is hypothesized that dimensionality reduction is how our brain deals with the dimensionality problem (Santello et al., 2013; Leo et al., 2016; Beyeler et al., 2019). The PCA-based approach has been proven to be a potentially useful tool for myoelectric prosthetic control.

Although computationally inexpensive, PCA has a single main feature, which is, arguably, its main limitation. Being a *linear* method, PCA can only account for linear relationships in the input data

and does not have a way of dealing with nonlinearities. However, biological signals are not purely linear, and one must consider this aspect if a more accurate representation of the original input data within the latent space is of desire.

One effective tool of *nonlinear* dimensionality reduction that emerged through the advancements in ML algorithms is an autoencoder (AE), which has been developed in the 1990s (Sanger, 1989; Kramer, 1991), following the creation of backpropagation process that many ML algorithms rely on (Rumelhart et al., 1985; LeCun et al., 1988). Being a type of neural network, AEs have the benefit of flexible structures, with varying numbers of layers, type of activation functions, and even incorporating a temporal component to account for the time domain the input data (Cho et al., 2014; Sutskever et al., 2014). As a result of their flexibility, they may be advantageous when encoding complex biological data for the purpose of low-dimensional control and could potentially outperform their linear counterparts, such as PCA.

During my PhD project, I studied and compared the performance of linear and nonlinear dimensionality-reduction methods, such as PCA and various AEs, in their ability to compress high-dimensional biological data. I then focused on developing a low-dimensional myoelectric controller for a high-dimensional virtual hand and studying what aspects of training affect user's ability to learn to operate the novel controller. Lastly, I explored what makes a latent space derived from EMG signals useful for control purposes and what properties of the input data might affect its structure.

Chapter 2 begins with the long-standing debate of the appropriateness of linear over nonlinear dimensionality-reduction techniques in the context of hand kinematics during grasping and American Sign Language (ASL) gestures. It addresses the statement in Yan *et al.* that “nonlinear

dimensionality reduction does not capture behaviorally relevant aspects of the hand kinematics more efficiently than does PCA” (Yan et al., 2020) and rebuts the claim by applying a simple nonlinear AE network to the given data and showing its superior performance in reconstruction over PCA.

Chapter 3 takes a deeper dive into the “linear vs nonlinear” issue by applying PCA and a simple nonlinear AE network to the newly collected hand kinematics data during ASL gestures, object grasps, and activities of daily living (ADLs). Furthermore, it explores the superiority of the nonlinear AE over PCA beyond the reconstruction power and looks at factors such as class separability and latent dimension variance in the context of a potential use for myoelectric control.

Chapter 4 revisits the debate on linear and nonlinear dimensionality-reduction techniques in the context of human gait and investigates the use of Long-Short-Term-Memory (LSTM) AEs to account for the temporal aspect of the input data. It also studies how efficiently the three methods (PCA, AE, and LSTM AE) are able to perform movement classification and individual identification – complex tasks that might be useful for prosthetic control purposes.

Chapter 5 focuses on the development of a low-dimensional controller, in which the decoder part of the AE network trained on hand kinematics data is used to operate a high-dimensional (17D) virtual hand via a low-dimensional (2D) space. In addition, in its three studies, it explores the aspects that might aid or inhibit learning of such a controller. The explored aspects were exploration time, difficulty due to the myoelectric interface, and various training paradigms to either implicitly or explicitly train the user on the connection between the underlying control space dimensionality and the dimensionality of the presented hand-controlling task.

Chapter 6 explores the application of various dimensionality-reduction techniques on myoelectric data in the context of a low-dimensional controller. There, it aims to bridge the gap of dimensionality mismatch in prosthetic systems not only from the kinematics side but also from the side of myoelectric signals. The optimality of the latent space for control purposes is assessed using three proposed factors: class separability, location of resting EMG with respect to the rest of the data, and spread of latent dimension variance. In addition, various properties of the input data are explored and their effect on the defined latent space factors is determined. Lastly, the chapter investigates factor stability for the AE networks across different model training.

Lastly, **Chapter 7** ties our findings into a final discussion and closing remarks and possible directions for future research are explored.

Chapter 2 The Long-Standing Debate of Linear vs Nonlinear Dimensionality-Reduction Methods¹

This letter is a commentary on the paper on “Unexpected complexity of everyday manual behaviors”, recently published in *Nature Communications* (Yan et al., 2020). We agree with the authors’ conclusions regarding the residual of dimensionality-reduction methods not representing mere noise and having a structure that is informative of the task. However, available theoretical and empirical evidence, including from our own work, is inconsistent with the article’s statement that “nonlinear dimensionality reduction does not capture behaviorally relevant aspects of the hand kinematics more efficiently than does [principal component analysis] PCA”.¹

Nonlinear dimensionality-reduction methods are many and diverse; nonlinear PCA (NLPCA) and Isomap, the two nonlinear methods used in the Yan *et al.* paper, are just two out of a big plethora of possibilities. We argue that based on these two methods alone, one cannot generalize about the inability of nonlinear methods to outperform their linear counterparts as stated by Yan *et al.* (Yan et al., 2020).

More specifically, in May 2020, we published a paper where we used two methods, PCA (linear) and an Autoencoder (AE, nonlinear) network to reduce the dimension of complex hand kinematics performed during American Sign Language gesturing, object grasping, and activities of daily living (Portnova-Fahreeva et al., 2020). In this work, we compared the performances of these two dimensionality-reduction techniques and expanded the earlier hand synergies work by Santello

¹ The content of this chapter was submitted as Portnova-Fahreeva, AA, Rizzoglio, F, Nisky, I, Casadio, M, Mussa-Ivaldi, FA and Rombokas, E. 2020. Matters Arising in Response to “Unexpected Complexity of Everyday Manual Behaviors” *Nature Communications*.

and Flanders (Santello et al., 1998). We observed and documented a clear advantage of the nonlinear AE method over PCA in capturing more variance of the hand kinematic with a smaller number of latent dimensions (*Figure 2-1*). A similar result was obtained by Tenenbaum *et al.* for image data (see *Figure 2* and related discussion in (Tenenbaum et al., 2000)) using Isomap, another nonlinear approach. They found that for a fixed number of latent dimensions, Isomap outperformed PCA in capturing complex information.

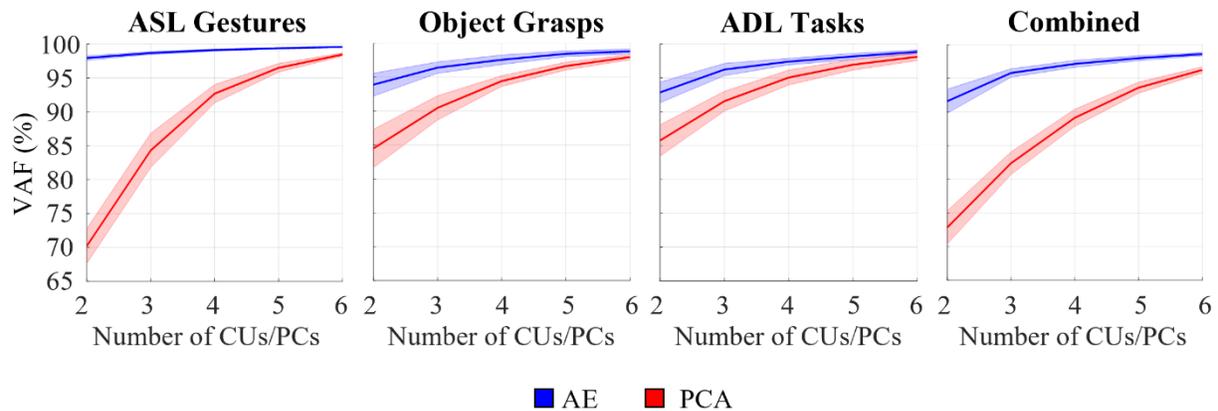


Figure 2-1. Performance of a nonlinear AE (blue) and PCA (red) when applied to complex hand kinematic datasets: American Sign Language (ASL) Gestures, Object Grasps, Activities of Daily Living (ADL) Tasks, and Combined data from the first three datasets (Portnova-Fahreeva et al., 2020).

We attempted to reproduce the results of Yan *et al.* by applying our nonlinear AE to their data and again observed that this nonlinear method clearly outperformed PCA (*Figure 2-2*).

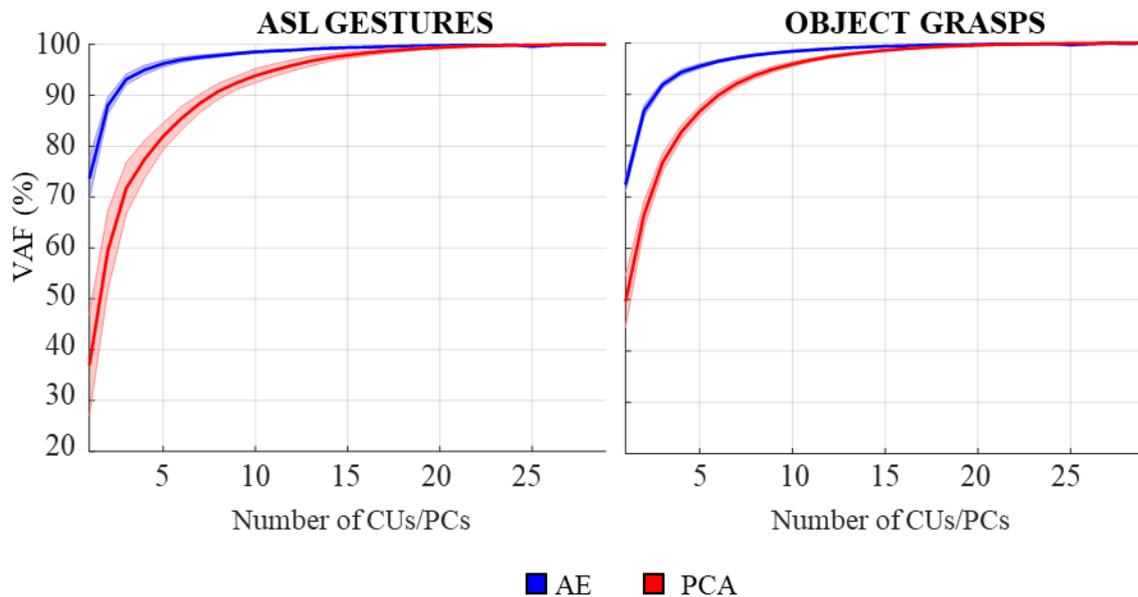


Figure 2-2. Performance of a nonlinear AE network (blue) and PCA (red) on American Sign Language (ASL) gestures and object grasps; data from Yan *et al.* (Yan et al., 2020).

Such discrepancies between the results of Yan *et al.* and our studies could depend on the type of nonlinear dimensionality-reduction network implemented in each case. The nonlinear PCA (NLPCA) method used in (Yan et al., 2020) is based on an AE similar to the one used in our study but nevertheless not identical. We used the simplest possible AE, which is based on minimizing the mean square error between the input pattern and the output pattern, a reconstruction of the input based on the latent variables (Kramer, 1991). In contrast, Yan *et al.* (Yan et al., 2020) used a variation of the AE that incorporates an additional constraint on the cost function of the network by forcing a hierarchization of latent space dimensions based on their variance.

For a dimensionality-reduction task, this hierarchical approach imposes additional constraints that limit its capacity. This discrepancy explains the difference in performance: the AE performs better than the NLPCA. In the two figures presented here, we have compared the AE results to those obtained with PCA. Unfortunately, we have been unable to apply the NLPCA method to our

original data since insufficient information was available about the specific architecture of the Autoencoder used in (Yan et al., 2020) to implement NLPCA. Our attempts to apply the NLPCA code with a default structure did not allow us to replicate results in Yan *et al.*, as the performance of the Autoencoder was heavily dependent on weight initialization. Perhaps, additional information on the architecture of the Autoencoder used to implement NLPCA would allow us to stably replicate the results in Yan *et al.* (Yan et al., 2020).

We also note that the hierarchical Autoencoder achieved a maximum performance of approximately 90% VAF (*Fig. 6* from (Yan et al., 2020)) instead of the expected 100%. This strongly supports our concerns that the hierarchical Autoencoder – at least in the particular implementation used by Yan *et al.* – could have been unable to reach a globally optimal solution during training. This discrepancy could also explain why the hierarchical AE in (Yan et al., 2020) exhibited less reconstruction power than PCA.

This result highlights that the choice of architecture of the nonlinear AE can strongly affect its performance and supports our concern about the use of the hierarchical approach as a benchmark for nonlinear dimensionality reduction.

In summary, we are concerned that the statement on the superior performance of linear methods over nonlinear dimensionality-reduction methods is incorrect as it does not represent the entire spectrum of possibilities offered by nonlinear methods. Without disputing the data in Yan *et al.* (Yan et al., 2020), the question that remains is whether nonlinear approaches do not capture relevant aspects of hand kinematics more efficiently than PCA or whether the NLPCA results just reflect limitations of this particular algorithmic approach.

To finalize, an additional question of importance when comparing different dimensionality-reduction approaches: what does it mean to be “more efficient”? Our previous study and the results of applying a straightforward AE to the data in Yan et al. indicate that the nonlinear AE clearly outperforms the linear PCA in terms of signal reconstruction. However, there might be other criteria such as the need for superposition and orthogonality of the elemental kinematic patterns, that would make linear methods of higher preference for dimensionality reduction. We thank you for your time in reading our commentary, submitted in the hope it can facilitate a productive discussion on the topic of dimensionality-reduction approaches and their performances.

Chapter 3 Linear and Nonlinear Dimensionality-Reduction Techniques on Full Hand Kinematics²

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 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.

² The content of this chapter has been published as Portnova-Fahreeva, AA, Rizzoglio, F, Nisky, I, Casadio, M, Mussa-Ivaldi, FA and Rombokas, E, 2020. Linear and Non-linear Dimensionality-Reduction Techniques on Full Hand Kinematics. *Frontiers in Bioengineering and Biotechnology*.

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 (Santello et al., 1998; Todorov and Ghahramani, 2004; Ingram et al., 2008). 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; Della Santina et al., 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., 2010; Matrone et al., 2012; Segil, 2013; 2015; Segil and Huddle, 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 (AE) networks. AEs 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. AEs provide an unsupervised method, reconstructing inputs in their outputs, without the requirement of labeled data. Most importantly, AEs 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 AE to that of PCA on examples of hand kinematics observed in human participants. In addition, it evaluates the case for the potential use of AE over PCA in a prosthetic controller.

Materials and Method

Experimental Setup

Ten unimpaired right-handed individuals (six males, four females, 32.8 ± 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 (*Virtual Motion Labs, Dallas, TX, USA*). A total of 20 signals were extracted from the glove that accounted for finger joint kinematics (*Figure 3-1*). The signals were recorded at a sampling rate of 100Hz.



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

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 3-1. 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

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.

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 3-2*) as described in the following sections.



Figure 3-2. Study setup consisting of three different phases: (A) American Sign Language (ASL) Gestures; (B) Object Grasps; (C) Activities of Daily Living (ADL) Tasks.

ASL Gestures

During the first phase of the experiment, the participants were asked to perform 10 different ASL gestures (*Figure 3-2A*). 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.

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 3-2B*) (Light et al., 2002). The testing board was placed 8cm 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 5s 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 5s 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 3-2C*). Each task was performed 10 times. The participants were given 7s to perform each task with a 5s 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 8cm 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.

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 AE.

Data Analysis

To study the effects of linear and non-linear dimensionality-reduction methods, we compared the performance of PCA and AE 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 AE was performed using TensorFlow, a Python (*Python Software Foundation, DE, USA*) library for machine learning applications developed by Google Brain (Abadi et al., 2016).

Nonlinear Autoencoder Architecture

The basic AE structure used for this experiment included a total of three hidden layers, the middle one being the bottleneck layer (*Figure 3-3*). Similarly to the original AE 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 (Kramer, 1991). 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 Y through the encoder (Equations 1, 2) and decoder (Equations 3, 4) parts of AE are:

$$layer_1 = \tanh (X * w_1 + b_1) \quad (1)$$

$$layer_2 = layer_1 * w_2 + b_2 \quad (2)$$

$$layer_3 = \tanh (layer_2 * w_3 + b_3) \quad (3)$$

$$Y = layer_3 * w_4 + b_4 \quad (4)$$

where w_i were the weights and b_i were the biases found during network modeling.

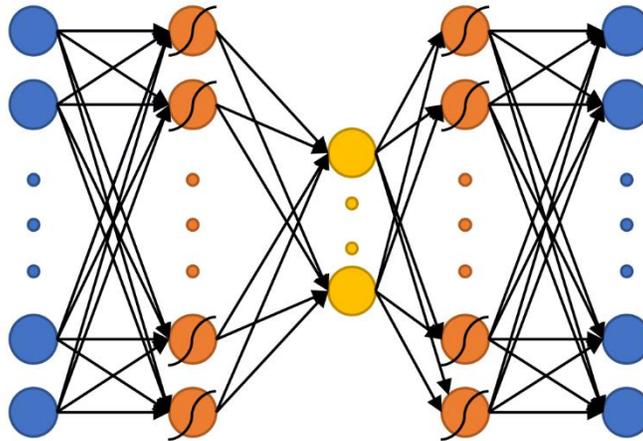


Figure 3-3. The non-linear Autoencoder (AE) network structure used in this study. Curved lines over neurons represent that non-linear activation functions (*i.e.*, hyperbolic tangent) were used to calculate that layer. Otherwise, activation functions were linear.

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 AEs, can be optimized further for improved performance. A simple AE 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 3-4*). 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).

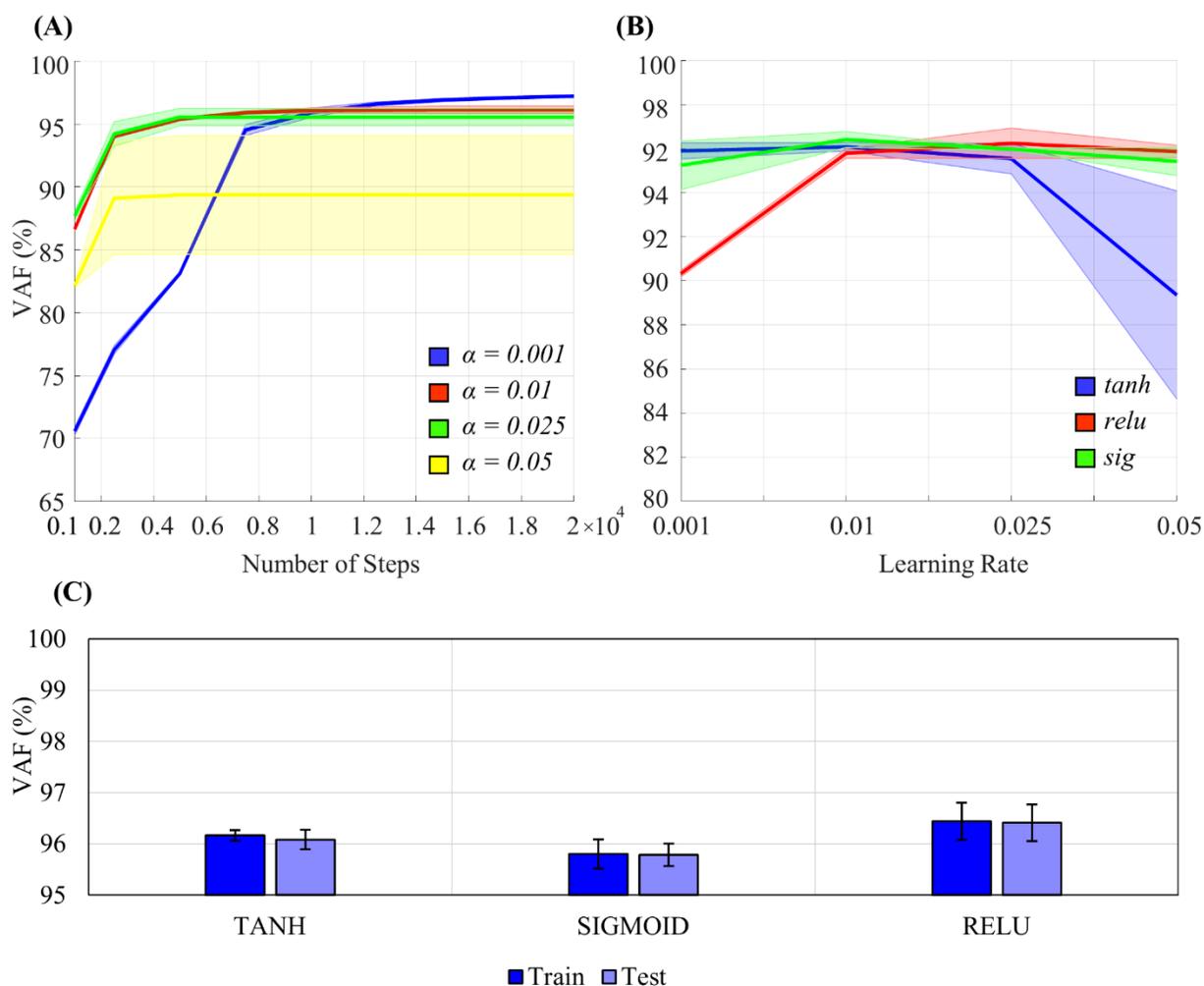


Figure 3-4. Effects of certain hyperparameters on the non-linear Autoencoder (AE) performance: **(A)** number of steps and learning rate (0.001,0.01,0.025,0.05); **(B)** learning rate and type of non-linear activation function (*tanh*, *relu*, *sig*). The performance is calculated using Variance Accounted 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)** AE performance difference on training and testing datasets for different activation functions with no regularization. Error bars represent 95% confidence intervals. The AE 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.

Learning rates of 0.01 and 0.025 produced the most stable results across the variety of number of steps tested (*Figure 3-4A*). 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 AE 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 AE. 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 3-4B*). As a result, we were free to choose any non-linear activation function for the AE structure, and the hyperbolic tangent was chosen for this experiment.

Adaptive Moment Estimation (ADAM) optimizer was used to speed up the training of the AE (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 3-4C*).

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

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 5*). A VAF of 100% indicated that the output and the input were identical.

$$VAF(\%) = \left(1 - \frac{\text{var}(Y - \hat{Y})}{\text{var}(Y)}\right) * 100 \quad (5)$$

Y —original data

\hat{Y} —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 AE.

Lastly, to explore the reconstructing performance of AE 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 AE. 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 6).

$$D = eig(cov(X)) \quad (6)$$

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 7).

$$PC_i = diag(D(:, i)) \quad (7)$$

For AE, latent dimensions were calculated by passing the normalized data X through the encoder part of the network (Equations 1, 2). CUs were the corresponding columns of $layer_2$ (Equation 8).

$$CU_i = layer_2(:, i) \quad (8)$$

Each latent dimension (PCs or CUs) was represented by A_i , an $m \times n$ 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 9).

$$\mu = \frac{1}{m} \sum_{j=1}^m A_j \quad (9)$$

Afterwards, the variance of each latent dimension, v_i , was calculated (Equation 10).

$$v_i = \frac{1}{m-1} \sum_{j=1}^m \|A_j - \mu\|^2 \quad (10)$$

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 11).

$$v_{dim} = \frac{v_i}{\sum_{i=1}^n v_i} \quad (11)$$

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 12).

$$range_{dim,avg} = \frac{1}{n} \sum_{dim=1}^n \max(v_{dim}) - \min(v_{dim}) \quad (12)$$

Visualization of Latent Trajectories

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 2PCs. The PC pairs for each dataset and each participant were then plotted on a 2D surface where PC_1 represented the x -axis and PC_2 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 AE. After reducing the input data to the latent manifold, the pair of CUs for each dataset was plotted on a 2D linear surface where CU_1 represented the x -axis and CU_2 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 AE 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 AE 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 AE and PCA. We used a 5-fold CV to calculate the accuracy on each dataset.

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, 1945) 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 AE 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 AE 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).

Results

Dimensionality Reduction

For all four datasets, AE outperformed PCA by reconstructing the input data with higher VAF for two, three, four, five, and six latent dimensions (*Figure 3-5*). The average VAF with just 2CUs across all datasets was 94% for AE whereas it was 78% with 2PCs for PCA.

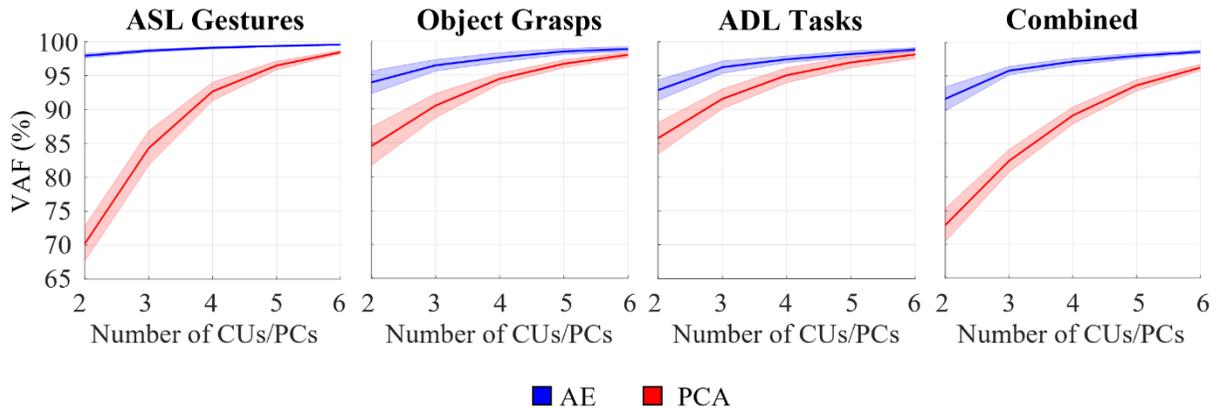


Figure 3-5. Performance of non-linear Autoencoder (AE, blue) vs. Principal Component Analysis (PCA, red) with 2, 3, 4, 5, and 6 dimensions in the bottleneck layer on four different datasets: American Sign Language (ASL) Gestures, Object Grasps, Activities of Daily Living (ADL) Tasks, and Combined. The 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. AE outperformed PCA across all datasets and for all number of latent dimensions tested in this study.

Wilcoxon Signed-Rank Test revealed an effect on VAF by the dimensionality-reduction method ($p < 0.001$). When comparing AE and PCA, the difference in the performance between the two methods decreased as the number of dimensions in the latent manifold increased (*Figure 3-5*).

By performing the Friedman's Test on VAF produced by PCA and AE 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 AE, 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 3-6A*). The R^2 values for a linear model between two variables were 0.58 and 0.54 for AE and PCA, respectively. The relationship is positive for AE and negative for PCA (observed from the slopes of the linear regression lines). Lastly, from *Figure 3-7*, one can observe the visual differences in the performance of two dimensionality-reduction methods. While AE (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.

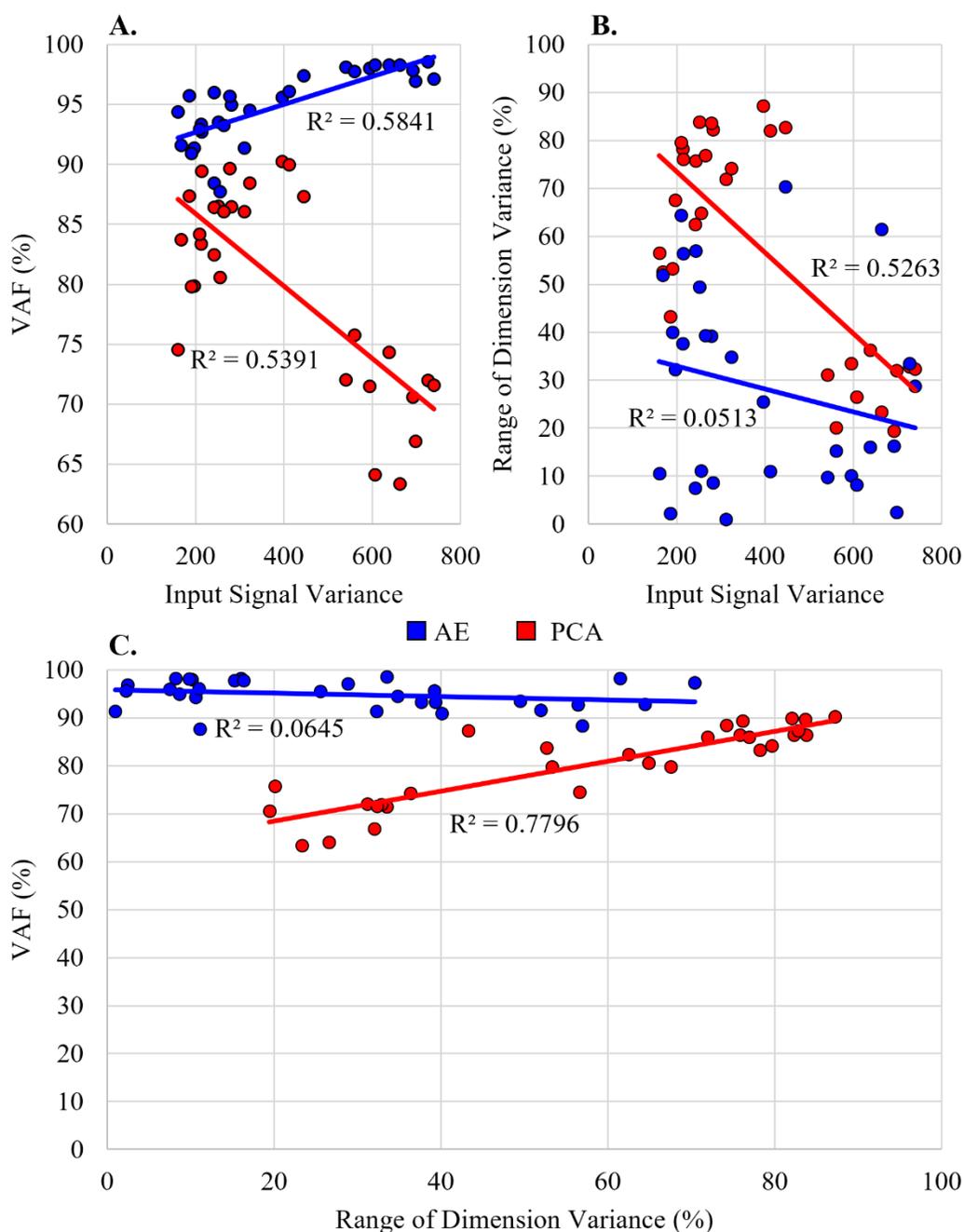


Figure 3-6. (A) Correlation between average signal variance of input data across 10 participants and Variance Accounted For (VAF) with two-dimensional latent manifold plotted for three different datasets (American Sign Language (ASL) Gestures, Object Grasps, Activities of Daily Living (ADL) Tasks) for non-linear Autoencoder (AE, 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 AE experienced no correlation.

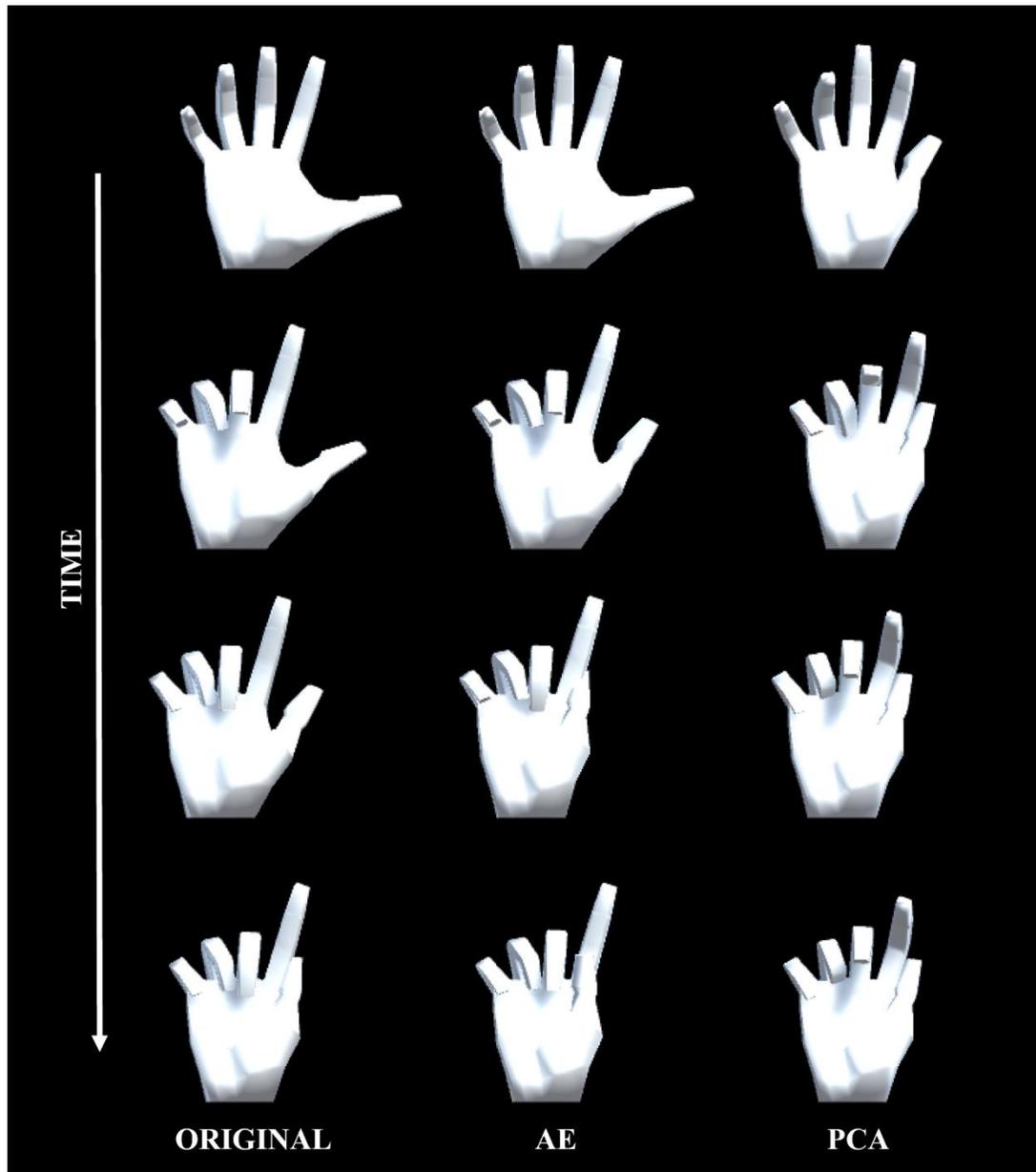


Figure 3-7. Visualization of gesture 1 from American Sign Language Gestures dataset, reconstructed from two-dimensional latent manifolds of non-linear Autoencoder (AE, middle column) and Principal Component Analysis (PCA, right column). The reconstructed gestures were

compared to original gesture (left column). A few snapshots of each gesture were taken in time from REST to ACTIVE states.

Dimension Variance

Variance spread across AE and PCA dimensions was plotted for participant P1 performing ASL Gestures (*Figure 3-8*). Variance dropped significantly for each new added PC dimension whereas AE 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.

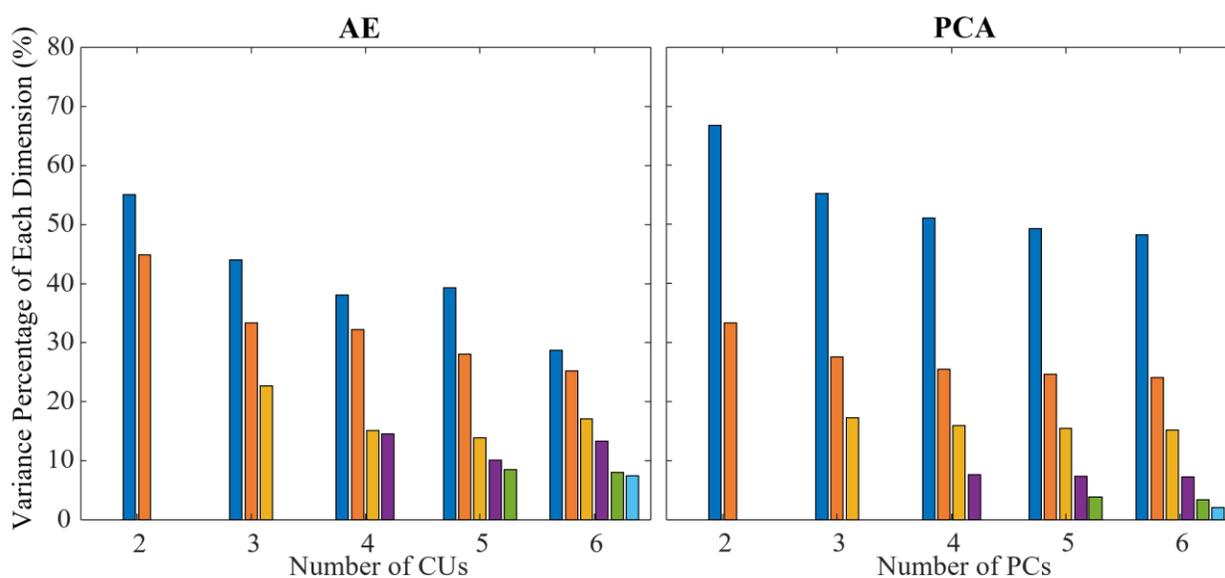


Figure 3-8. Variance percentage of each latent dimension (represented with different colors) with respect to the overall variance of the dimensions considered in the analysis [number of Principal Components (PCs)/Coding Units (CUs)] for P1 performing American Sign Language (ASL) Gestures. Two, three, four, five, and six dimensions in the bottleneck are compared for non-linear Autoencoder (AE) and Principal Component Analysis (PCA) and ranked by order of decreasing variance. AE exhibited smaller drop of variance across latent dimensions in comparison to PCA for P1.

Such behavior was consistent across all participants, as described by the average difference between dimensions with the highest and lowest variance (*Figure 3-9*). This difference was greater

for PCA than for AE 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.

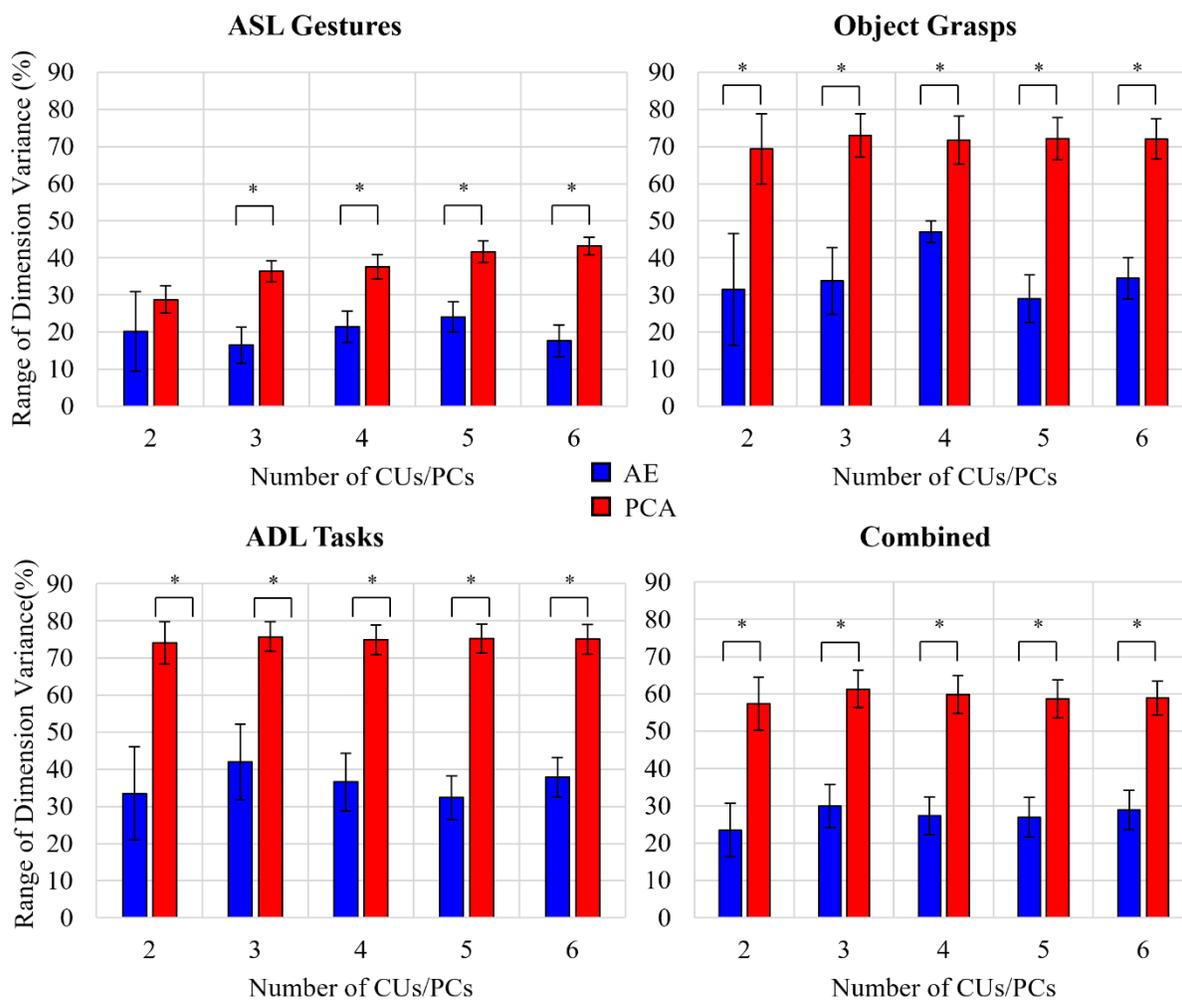


Figure 3-9. Difference between dimensions with highest and lowest variance values averaged across all participants for four datasets: American Sign Language (ASL) Gestures, Object Grasps, Activities of Daily Living (ADL) Tasks, and Combined. Statistical significance: $p = 0.106$ for ASL Gestures for 2CUs/PCs; *indicates statistical significance of $p = 0.002$. AE 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.

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 3-6B*). No such correlation was found for AE (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 ($R^2 = 0.78$) and no correlation for AE ($R^2 = 0.06$) (*Figure 3-6C*).

Visualization of Latent Trajectories

AE and PCA latent trajectories were visualized for 2CUs/PCs in the case of participant P1 performing ASL Gestures (*Figure 3-10A*), Object Grasps (*Figure 3-10B*), and ADL Tasks (*Figure 3-10C*). All trials used in the analysis were plotted, and separate gestures were indicated using different colors.

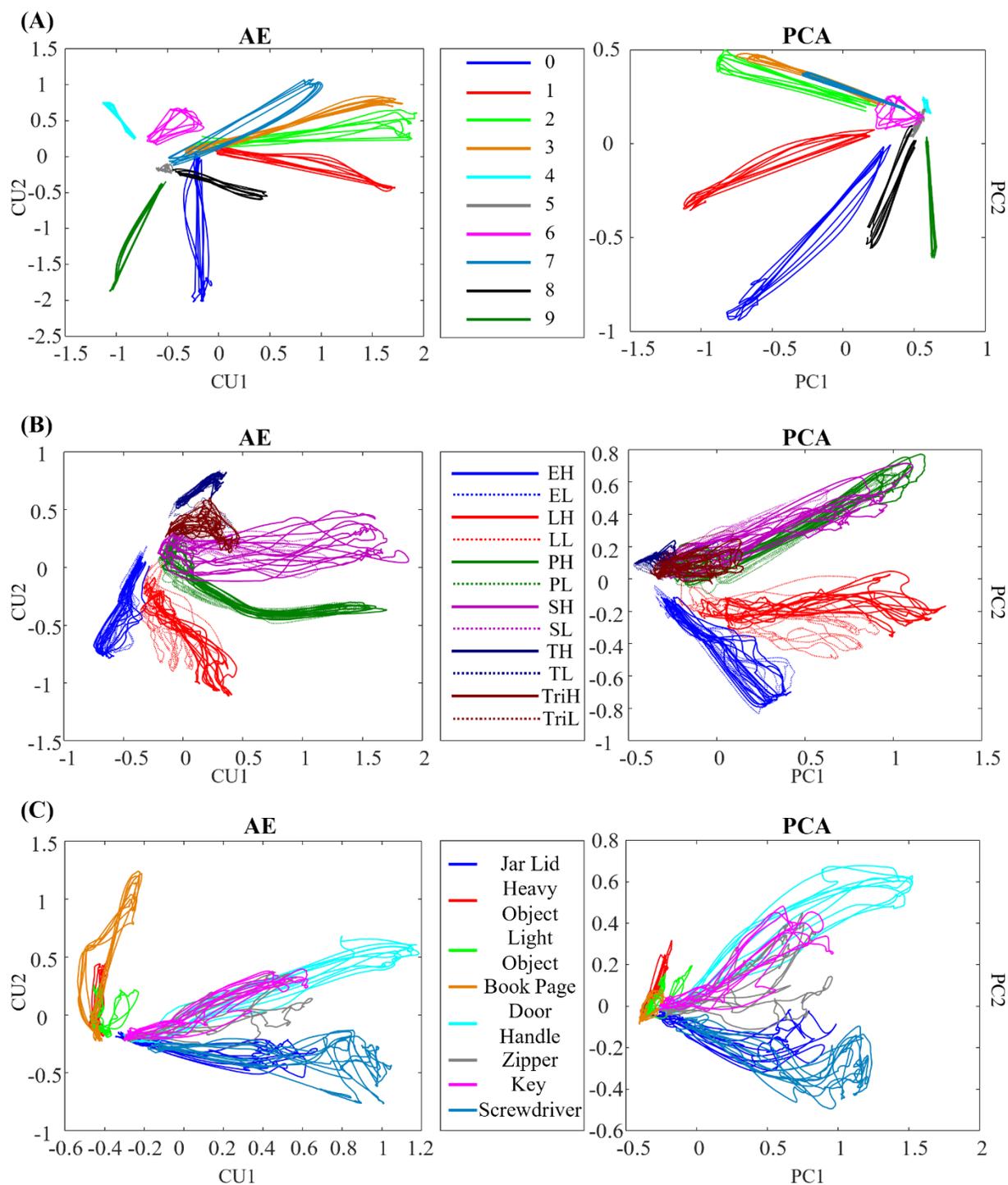


Figure 3-10. Visualization of the latent trajectories for AE and PCA for P1 performing (A) American Sign Language (ASL) Gestures, (B) Object Grasps, and (C) Activities of Daily Living (ADL) Tasks. The hand movements tested are represented with different colors. The latent

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).

The overall structures of the plotted latent trajectories were similar across PCA and AE, 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 AE 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.

Movement Separability

SoftMax regression on the 2D manifold of AE and PCA did not reveal any significant difference in separability of movements between the two methods (*Figure 3-11*). 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).

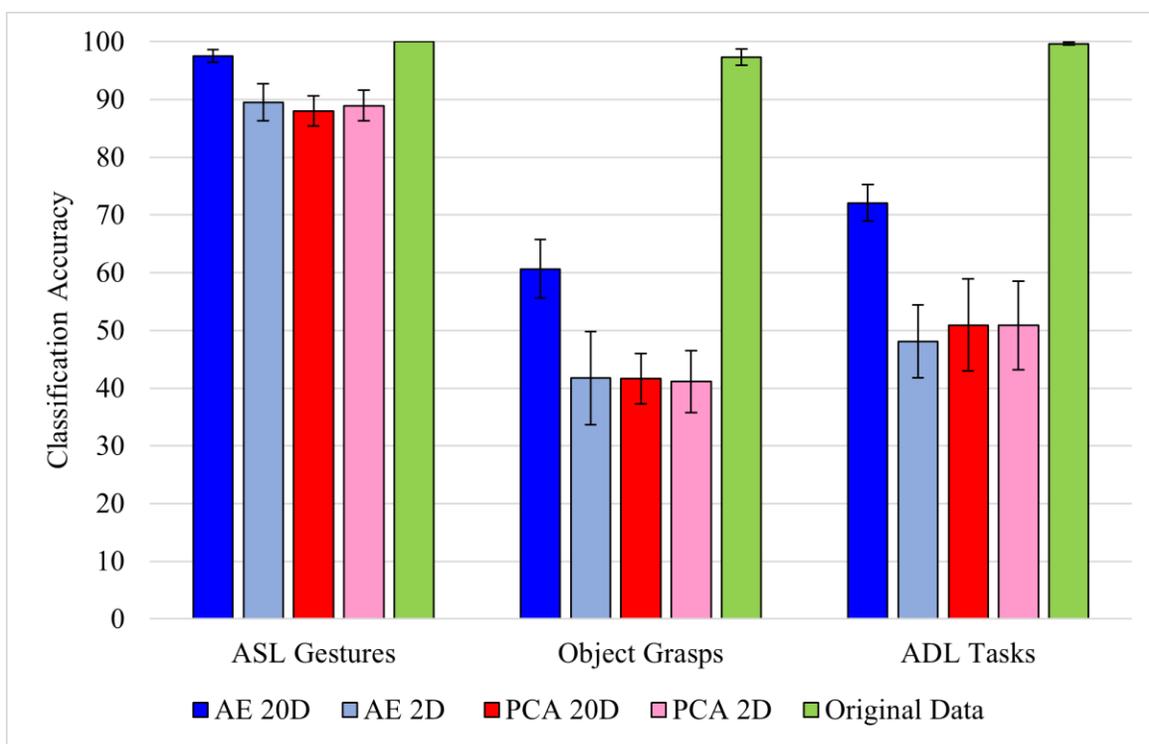


Figure 3-11. Accuracy of SoftMax regression applied to different datasets [American Sign Language (ASL) Gestures, Object Grasps, Activities of Daily Living (ADL) Tasks] across all participants. Regression was applied to original input data (green), reduced non-linear Autoencoder (AE) 2D (light blue) 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 AE 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 AE with the former being more separable than the latter.

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, AE 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 AE. The classification accuracy was at nearly 100% across all datasets (*Figure 3-11*).

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 3-11*).

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 study, a non-linear equivalent of PCA, AEs, 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 AE a potentially effective at supporting continuous posture control for prosthetic hands.

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 (Santello et al., 1998; Todorov and Ghahramani, 2004; Weiss and Flanders, 2004; Feldman and Latash, 2005; Ting and Macpherson, 2005).

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 2PCs were sufficient to account, on average, for ~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's (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 2PCs. Similar behavior could be observed in Todorov and Ghahramani's (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), AE 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 2CUs for AE. As a result, AE 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 AE outperformed PCA by reconstructing over 90% of data variability

with only 2CUs. 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 (Romero et al., 2010; 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 AE. 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 AE 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 AE when used as a control method for hand prostheses. Higher VAF value signifies that with a smaller number of control signals, the AE-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 AE 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), AEs 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.

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, AE 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 AE, 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, AE 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.

Visualization of Latent Trajectories

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 AE since CUs are neither orthogonal nor linear elements.

As one visualizes the latent trajectories of an AE 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 AE 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 AE 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.

Movement Separability

Separability between AE and PCA in 2D was participant- and movement-dependent, implying that some participants created a more separable AE 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 AE 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 AE 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 AE 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 3-10* 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 3-10* also shows that there is not a difference in classification accuracy between AE 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 3-5, 3-6A*).

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 AE-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 AE results must be made.

Although there was a significant improvement when going from 2D to 20D for AE, 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 AE. 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.

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 AE, 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 (3) and (4). In such way, the development of the hardware of the prosthetic hand would rely on the composition of the decoder component of AE.

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 AE will be explored with EMG signals in future research of improving myoelectric control of hand prosthesis.

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 (Hargrove et al., 2006; Scheme and Englehart, 2011; Young et al., 2011; Castellini et al., 2014; Atzori et al., 2016; Geethanjali, 2016).

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 AEs 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 (Muceli and Farina, 2011; Muceli et al., 2013; Ngeo et al., 2014; Geethanjali, 2016). 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 a less effective control tool for prostheses.

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

Chapter 4 Dimensionality Reduction of Human Gait for Prosthetic Control³

Introduction

Models of human gait are the foundation upon which lower limb prosthesis controllers are built. Because gait is highly complex and multidimensional, these models take advantage of simplifying assumptions to narrow the problem space. Early above-knee prostheses relied upon events during stance and swing phases to trigger locking and unlocking of a mechanical knee, such as the knee hyperextension moment at toe-off (Mauch, 1968). As prosthetic technology has advanced, so have the underlying models. Variable damping knees use on-board sensors to detect speed and phase, adjusting knee and ankle joint control parameters to mimic human gait (Highsmith et al., 2010). Today, powered prostheses that generate work during gait are gaining in popularity in research circles (Azocar et al., 2020). However, the challenge of controlling prostheses has been recently brought again to attention (Tucker et al., 2015; Iandolo et al., 2019), and only highlighted by the untapped potential of powered devices to restore mobility. We assert that generating useful representations of human movement is necessary to unlock the potential of such devices.

Gait models can be used to generate reference trajectories of kinematics or torque, or inform a set of control parameters for powered prostheses. Generating safe and reliable trajectories and parameters, given the complexity of human gait, poses a challenge. To do so, simplifications are made. At a high level, activities such as level ground walking, stair navigation, and ramp

³ The content of this chapter has been published as Boe, D, Portnova-Fahreeva, AA, Sharma, A Rai, V, Sie, A, Preechayasomboon, P and Rombokas, E, 2021. Dimensionality Reduction of Human Gait for Prosthetic Control. *Frontiers in Bioengineering and Biotechnology*.

navigation, can each be called an individual “mode” of movement. At a lower level, control is achieved with respect to phases of gait. For instance, Simon *et al.* split the gait cycle of each mode into finite states delineated by gait phases, in which each state corresponds to a set of impedance parameters, totaling 140 tunable parameters (Simon et al., 2014). However, adding additional modes creates more tuning parameters, which poses an additional challenge. More recent approaches have reduced the number of tuning parameters by creating unified gait models that span across modes. Quintero *et al.* has developed a gait model that generates knee and ankle reference trajectories with respect to speed, phase, and incline (Quintero et al., 2018). This significantly reduces the solution space while maintaining expressiveness of the model output. However, it remains invariable to idiosyncratic gait characteristics, which Quintero *et al.* also identifies as the largest source of variability.

Techniques to simplify gait can be used to address these challenges. One such technique is to reduce the dimensionality of gait by learning its “principal components” from real world data. Dimensionality reduction techniques like Principal Component Analysis (PCA) have been used to identify a variety of pathological gaits (Deluzio et al., 1997; Matsushima et al., 2017; Slijepcevic et al., 2017; Chen et al., 2020) and detect differences in kinetics with transfemoral amputation (Soares et al., 2016). Unlike standard PCA, nonlinear dimensionality reduction techniques like autoencoders are able to fit a nonlinear function to nonlinear data, though it is unclear which technique is suited for gait—which is a highly structured, periodic behavior. We have previously explored how PCA compares to an autoencoder for dimensionality reduction of hand kinematics, as it pertains to priorities for prosthetic control (Portnova-Fahreeva et al., 2020). In this study, we will present a similar analysis using lower limb kinematics collected during gait activities. We will

also compare performance between dimensionality reduction techniques on tasks relevant for prosthetic control - movement classification and individual identification.

Movement classification is a broad, albeit powerful, way to simplify gait. For many lower limb prostheses, selecting the desired movement class, like flat ground walking and stair walking, is typically performed via user input, such as bouncing on the heel three times. Requiring manual input from the user side steps the challenge of selecting the desired movement class using only sensor inputs. Well performing dimensionality reduction techniques may simplify this challenge, enabling a classifier to automate selection of a movement class, and thus minimizing the control burden placed on the user. However, within a single movement class, gait may be highly variable from one individual to another, either due to pathology, amputation, or idiosyncrasy. These variations are assumed to be chosen to optimize over some set of parameters, like stability (Herssens et al., 2020) or metabolic cost (Summerside et al., 2018) and so are important to preserve during dimensionality reduction.

Gait models and associated prosthesis control algorithms are also designed to be highly reliable. Because small, rare errors in the model or controller can have catastrophic consequences, simpler solutions are favored. Machine learning algorithms are capable of taking on large, high dimensional problems, but are prone to errors on unseen data and suffer from a lack of interpretability. However, significant interest in machine learning methods over the last decade have resulted in the creation of novel algorithms that offer unique potential for modeling gait. We have previously demonstrated the viability of using machine learning to predict joint kinematics for lower limb prosthesis control (Rai and Rombokas, 2019; Rai et al., 2020). In this study, we use autoencoders, which are a class of self-supervised networks with flexibility to handle virtually any

type of data. We will employ two autoencoders, one trained to reconstruct a single pose from gait (Pose-AE), and a recurrent autoencoder trained to reconstruct an entire movement (Move-AE).

In this study, we seek to better understand how gait data can be simplified using dimensionality reduction. In the first part of this study, we compare the dimensionality reduction performance of PCA on poses and an autoencoder on poses taken from lower limb gait data. In the second part, we will compare performance on movement and individual classification tasks of PCA on poses, an autoencoder on poses, and a recurrent autoencoder on movements. We expect that autoencoders will outperform PCA in all cases, as demonstrated on hand kinematics in our prior work (Portnova-Fahreeva et al., 2020).

Materials and Methods

Data Collection

Gait data was collected in a previous study (Rai and Rombokas, 2019). Participants wore the Xsens Awinda suit (*Xsens Technologies, Enschede, Netherlands*), a wearable motion capture suit consisting of 17 body-worn sensors. Xsens Analyse software processes raw sensor data to provide joint kinematics in a 3D environment. All angles are in a 1×3 Euler representation of the joint angle vector (x, y, z) in degrees, calculated using the Euler sequence *ZXY* using the International Society of Biomechanics standard joint angle coordinate system (Wu et al., 2002). Recruitment and human subject protocols were performed in accordance with the University of Washington Institutional Review Board approval and each subject provided informed consent. De-identified data can be made available, via a data use agreement, upon request to the authors.

From this dataset, we are examining 10 participants who performed flat ground walking and 14 participants who performed stair ascent and descent. All participants in all groups were unique.

Flat ground data consists of participants walking at a self-selected speed down a long public corridor. Stair data consists of participants repeatedly descending a wide public 13-step staircase, turning around at the landing, and ascending the same staircase, also at a self-selected speed.

We are also using the Virginia Tech Natural Motion Dataset (Geissinger and Asbeck, 2020), also collected using an Xsens system. It contains 40h of natural, unscripted movement from 17 participants, including 13 participants on a college campus and four participants working in a home improvement store. This dataset is representative of movement in daily life, as compared to constrained activities like steady state forward gait.

Data Processing

Xsens features a real-time engine that processes raw sensor data for each frame and algorithmically fits a human body model to estimate anthropomorphic joint and segment data. A post processing engine includes information from the past, present, and future to get an optimal estimate of the position and orientation of each segment. This “HD” processing raises the data quality by extracting more information from larger time windows and modeling for skin artifacts, etc. but also takes significantly longer time. We used HD processed data as training data for all three datasets.

Each dataset was standardized according to the aggregated statistics of all three datasets. Three lower limb joints on each side of the body (hip, knee, ankle) were chosen for analysis. Each joint can be represented in frontal, transverse, and sagittal planes. Frontal and transverse plane motion was dropped for the knees, due to its propensity to reflect sensor noise over meaningful physiological movement. Because the natural motion dataset contains long periods of inactivity, such as sitting at a desk, it was filtered by pelvic velocity such that only moments punctuated by

movement of the pelvis would be included. The subjects of each dataset were then allocated into either train or test sets for analysis. For a comparison study of different techniques, achieving high performing, generalizable results are not our primary aim. Rather, we would like to highlight the attributes of how these techniques interact with the data without optimization. For this reason, we only consider the training set throughout the rest of this study (*Table 1*).

TABLE 4-1. Details of the training dataset used in this study. Additional individuals were held out for a future testing dataset. Sampled duration reflects the combined length of the recordings from which samples were uniformly extracted.

Dataset	n	Male/Female	Age (yrs)	Height (cm)	Samples	Recording Duration
Flat Ground	8	4/4	26.2 ± 2.7	174 ± 10.9	38,196	1 h 38 min
Stair Walking	11	8/3	24.7 ± 3.5	173 ± 11.2	54,755	2 h 29 min
Natural Movements	13	14/3	20-58	179 ± 7.3	34,779	1 h 36 min

Data Analysis

Data analysis was performed in Python 3.7 and the machine learning was implemented using *Tensorflow 2.0* using a single GPU. Visualization of lower limb poses was achieved using an open-source humanoid model in Unity (*Unity Technologies, San Francisco, CA, United States*).

Principal Component Analysis

PCA was performed for each dataset using the respective covariance matrix. PCA achieves dimensionality reduction by projecting the original data by the space defined by its principal components (PC), each of which are vectors aligned to maximally capture remaining variation in the data. A limited number of principal components often explain the majority of variation in the data, resulting in a lower dimensional space than the original data. This space will be referred to as the latent space.

Pose Autoencoder

The autoencoder is one variant of the encoder-decoder architecture. Notably, encoder-decoder architectures have been used to power breakthroughs in natural language processing (Devlin et al., 2018), but have been applied to computer vision (Hossain et al., 2019), time series analysis (Lim and Zohren, 2021), and human movement (Petee et al., 2019). Petee *et al.* used such techniques to produce manifolds of human dancing, from which samples of novel dance choreography may be decoded. Geissinger and Asbeck utilized similar principles to infer complete joint information from sparse sensor input on the natural motion dataset considered here (Geissinger and Asbeck, 2020).

Critically, the autoencoder contains a bottleneck through which it is forced to learn features of the data. The activations of the bottleneck layer represent the data in the latent space. Nonlinear activation functions in each layer can capture nonlinear relationships in the data, though often (but not always) at the cost of interpretability.

As illustrated in *Figure 4-1*, inputs to Pose-AE were of size 1×14 and consisted of hip, knee, and ankle joint angles, as described in the Data Processing section. For each time series of joint angles, inputs were sampled every 0.166s. This 1×14 vector is then passed through the encoder, after which it can be represented by a 1×2 vector in the latent space. The decoder then attempts to reconstruct the original 1×14 vector from the 1×2 latent vector. The reconstruction error between the decoded 1×14 vector and the input 1×14 vector backpropagate through the network layers, forcing the network to learn how to best represent the 1×14 input vector as a 1×2 latent vector. In other words, Pose-AE was trained to reconstruct 14 dimensional “snapshots” of lower limbs from only two dimensions.

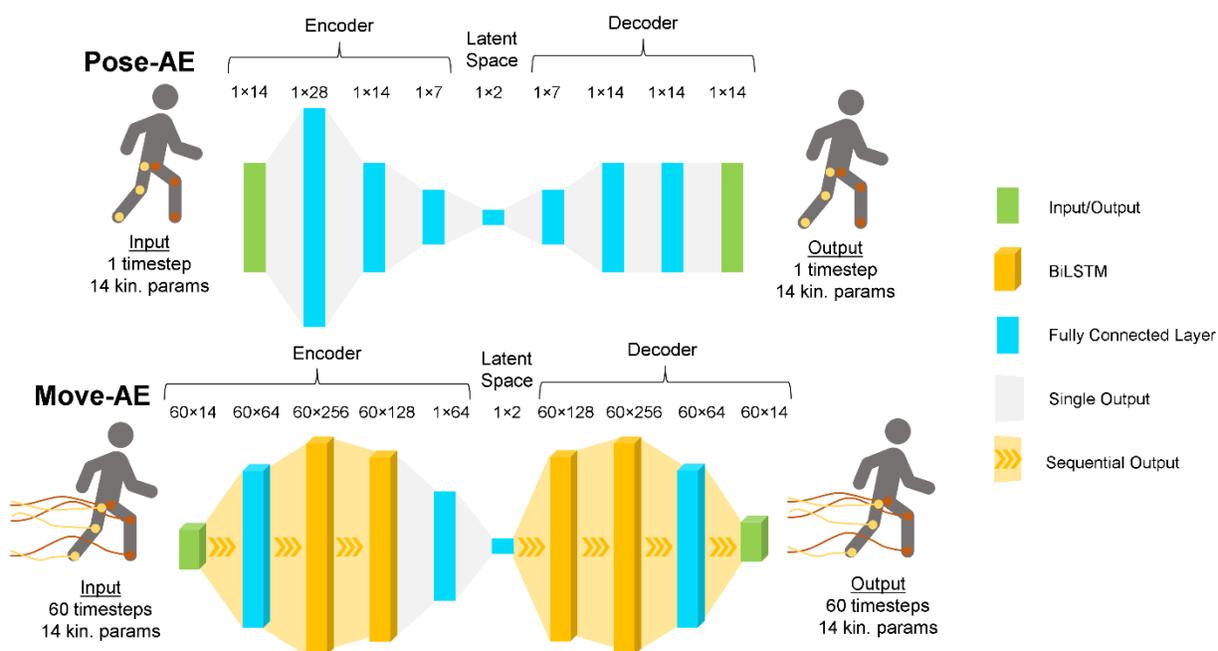


Figure 4-1. Architecture for the Pose-AE network and Move-AE network. Both networks exhibit the classic autoencoder bottleneck shape. Whereas Pose-AE takes single poses as input, Move-AE makes use of stacked recurrent bidirectional LSTM layers to accept entire movement trajectories as input. Both networks embed each input as a single point in a two-dimensional latent space.

Hyperparameter optimization was performed using a single random participant from the flat ground training dataset, with a training and validation split of 50/50. Tuning on a single subject was done as an alternative to k-fold hyperparameter tuning, which becomes combinatorially expensive with three activities. Considering the aim of the study is to compare techniques, not seek maximal performance, the authors decided to err on the side of underfitting, to ensure the most fair comparison across techniques. Hyperparameter choices were found to be insensitive to the chosen subject. We also tested several network widths and depths and found the best results with a three-layer block for both the encoder and decoder (*Figure 4-1*). Batch normalization was implemented in the encoder to mitigate overfitting. Each configuration was evaluated by its reconstruction loss. Adaptive Moment Estimation was used to optimize learning during training. All Pose-AE models

were trained using full batch gradient descent for 8,000 iterations, which was heuristically determined to achieve model convergence before showing evidence of overfitting.

Anecdotally, we discovered very little sensitivity of hyperparameters by subject.

Movement Autoencoder

Unlike Pose-AE, Move-AE reduces entire movements. The input to Move-AE was a one second window of all 14 joint kinematics, thus a sequence length of 60 timesteps, given the original 60Hz recording rate. Though this time series data is higher dimensional, the signal exhibits both autocorrelation and periodicity, making the problem more tractable. Indeed, the fact remains that if one was asked to draw a canonical joint trajectory during flat ground gait of any length, only two pieces of information are required to adequately represent it: cadence and phase. Recurrent layers like the Long Short Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) are capable of extracting key information from time series data, and Bidirectional LSTMs (Schuster and Paliwal, 1997) have been employed here (*Figure 4-1*). A point in the latent space now represents one second of movement, rather than a snapshot of a pose. We chose one second of movement as a sufficient length of time to capture the context of a given pose. This is in contrast to the pose autoencoder and PCA, which would be unable to determine if one was walking forward or backward because they cannot learn the temporal dependencies within movements.

Inputs to Move-AE consisted of the same 14 joints and planes as Pose-AE, but now extend to include 1 s of movement at 60Hz. These inputs are similarly sampled every 0.166s, regardless of gait phase, meaning there is overlapping data between multiple inputs. The input vector is now shaped 60×14 (60 timesteps, 14 joints and planes). As before, the 60×14 input vector passes through the encoder, after which it is represented as a 1×2 latent vector. The decoder

reconstructs the entire 60×14 input from this 1×2 latent vector. Move-AE learns to represent entire movement trajectories of the lower limbs as a 1×2 vector.

Hyperparameter optimization was performed as previously described. The best performing architecture was found to have two bidirectional LSTM (biLSTM) layers for both the encoder and decoder and time distributed fully connected layers before and after. Making a fully connected layer “time distributed” allows it to accept sequential data by passing each timestep through individually. Time distributed fully connected layers of width 64 are used to generate the input sequence to the encoder biLSTM block, and generate the output sequence from the decoder biLSTM block. An intermediate fully connected layer was included after the encoder biLSTM block to facilitate dimensionality reduction to two dimensions in the latent layer. All models were trained for 8,000 iterations as previously described, except we used mini batch gradient descent with a batch size of 32 to decrease training time with larger inputs.

Variance Metrics

As described in our previous work (Portnova-Fahreeva et al., 2020), Variance Accounted For (VAF) is a measure of how well a model reconstructs an input from the latent space. A VAF of 100% indicates the reconstructed output is identical to the input. VAF was evaluated for inputs reconstructed by PCA and autoencoder for each dataset. The equation is presented again here for clarity (*Equation 1*).

$$VAF(\%) = \left(1 - \frac{\text{var}(Y - \hat{Y})}{\text{var}(Y)}\right) * 100 \quad (1)$$

We compare how variance is distributed between each dimension of the latent space. Principal components were ranked by variance explained and normalized to the sum of previous principal

components, thus converting the variance of each PC into a ratio of the total variance. The autoencoder was trained using various bottleneck widths. Similarly, the variance of activations in the bottleneck were normalized to the sum of all variances in the bottleneck layer. As per previous results (Portnova-Fahreeva et al., 2020), dimensional variance is expected to be more uniformly distributed in the autoencoder, which does not have the constraint of PCA’s orthonormality, but is capable of sharing variance across multiple latent dimensions. We also report the Root Mean Square Error (RMSE) between the original input and the reconstructed input. Note how RMSE differs from VAF, in that it directly measures error in reconstruction, whereas VAF measures what proportion of variance has been captured.

Classification Tasks

We compare the performance of all three dimensionality reduction methods on two classification tasks: movement classification and individual identification. For both tasks, we use a Support Vector Machine to classify within the latent space, performed using the scikit-learn (Pedregosa et al., 2011) implementation, in turn based on the formulation presented here (Chang and Lin, 2011). We use a radial basis function kernel to improve classification accuracy, given the low-dimensional latent space. All parameters were fixed for all tasks and latent spaces.

The movement classifier sought to determine whether a given point in the latent space represented flat ground walking or stairs navigation. The natural movement dataset was excluded due to the presence of both activities within the single dataset. The individual classifier sought to identify the individual from which a given input in the flat ground walking dataset originated. The error was calculated as the number of erroneously classified inputs divided by the total number of inputs. In both cases, the training dataset was used, and training was repeated 10 times for each model to

capture a better range of outcomes. The Kolmogorov-Smirnov (KS) test was employed to test if Pose-AE and Move-AE classification errors were drawn from different distributions. The KS test is well suited when the sets under comparison exhibit different variances.

The classification results indicate how separable different movements and individuals are in the latent space. High separability will result in high classification accuracy, indicating that the dimensionality reduction technique has preserved high amount of information about the input. This test also allows a direct comparison to be made across techniques as diverse as PCA, autoencoders, and recurrent time-sensitive autoencoders.

Results

Dimensionality Reduction

Pose-AE exhibited better pose reconstruction than PCA for flat ground and stair walking, but neither Pose-AE nor PCA was able to adequately reconstruct natural poses from a two-dimensional latent space (RMSE 0.63 vs 0.55, 0.80 vs 0.71, 1.05 vs 1.02) (*Figure 4-2*). For flat ground and stair walking, visual inspection of randomly chosen reconstructed poses by each method illustrate how even small improvements in RMSE may result in qualitatively improved pose reconstruction, especially in regards to sagittal plane. However, both methods perform poorly on the natural movement dataset. An RMSE > 1 indicates neither method is an improvement from simply reconstructing the mean pose.

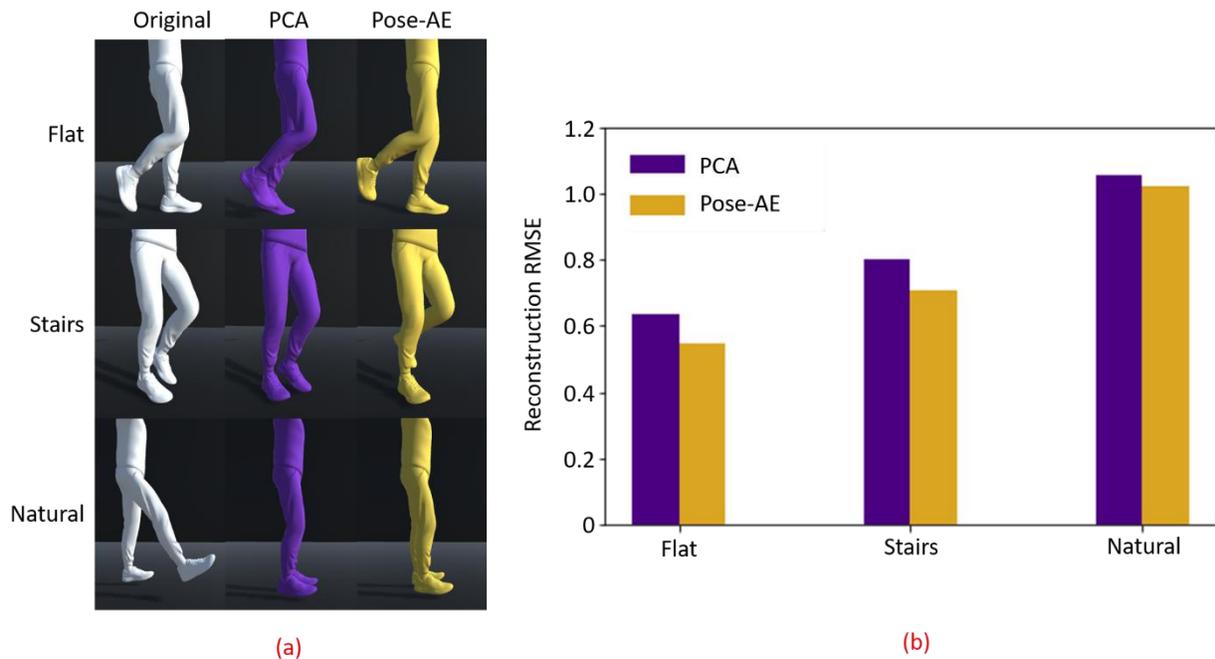


Figure 4-2. (A) For each dataset, a randomly chosen sample was reconstructed using PCA and Pose-AE. Pose-AE produced qualitatively improved poses over PCA, though neither was able to reconstruct poses from natural movements dataset, instead electing to reconstruct a mean standing pose. (B) RMSE of joint angles was calculated for each dataset and method. Pose-AE shows a modest improvement over PCA.

Similar to previous findings (Portnova-Fahreeva et al., 2020), Pose-AE captures greater variance in the data than PCA during dimensionality reduction, especially at low dimensions (*Figure 4-3*). Dimensional variance is more evenly distributed with Pose-AE than PCA. Though neither method was suited to reconstruct natural movement poses, Pose-AE retained an evenly distributed dimensional variance - indicating the capability to share dimensional variance across dimensions is inherent to autoencoders, regardless of dataset.

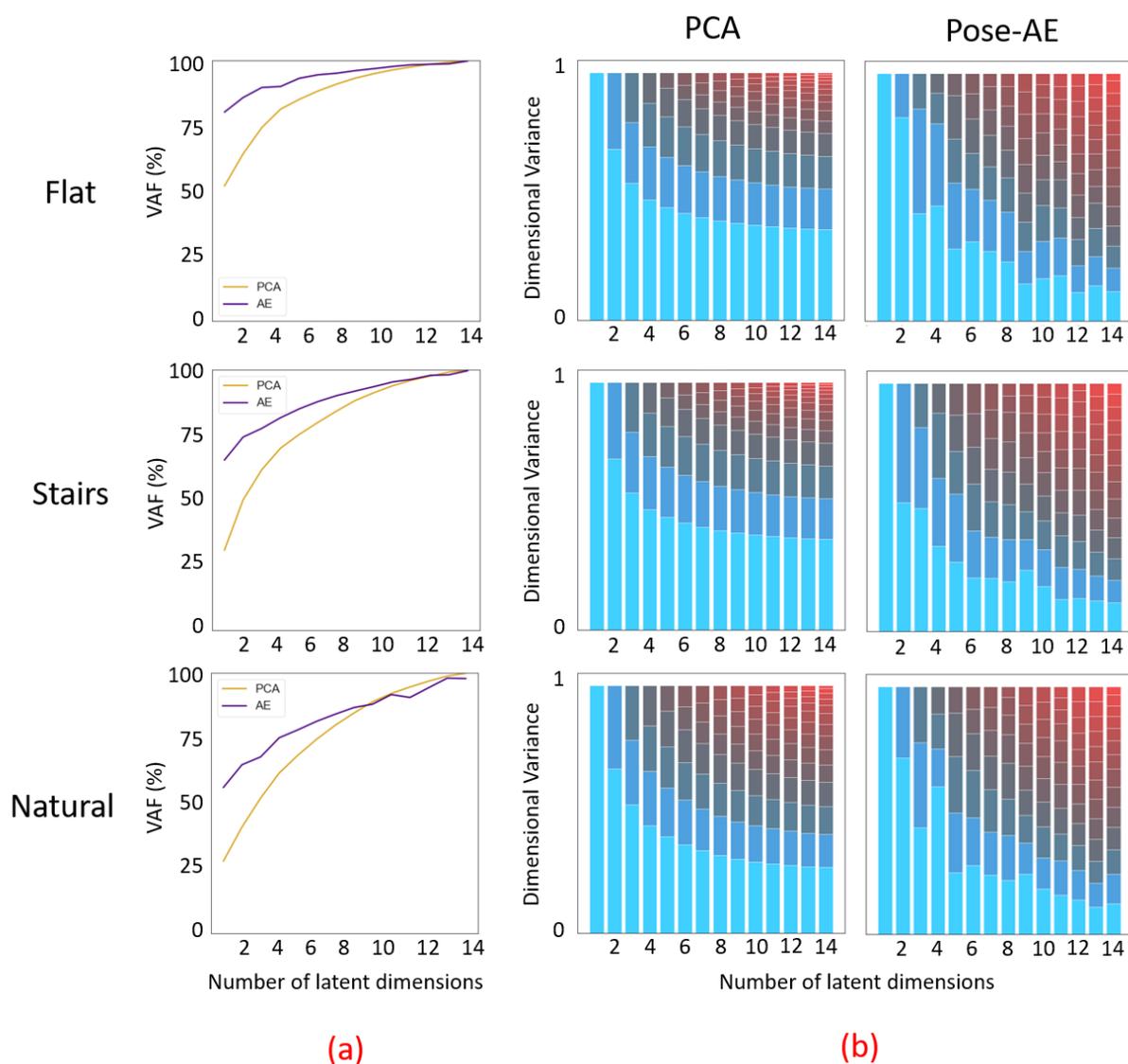


Figure 4-3. (A) For each dataset, a randomly chosen sample was reconstructed using PCA and Pose-AE. Pose-AE produced qualitatively improved poses over PCA, though neither was able to reconstruct poses from natural movements dataset, instead electing to reconstruct a mean standing pose. (B) RMSE of joint angles was calculated for each dataset and method. Pose-AE shows a modest improvement over PCA.

Movement Separability

Both autoencoder models produced latent spaces more suited for movement classification than PCA (error 21.8% PCA; $11.7 \pm 3.4\%$ Pose-AE; $3.3 \pm 2.0\%$ Move-AE). Move-AE exhibited

significantly different movement classification performance than Pose-AE (KS test, $p < 0.0001$). The latent spaces of each method's best performing model are visualized in *Figure 4-4*. Unlike PCA, Pose-AE was sensitive to differences in flat ground walking and stair walking, thus embedding them with little overlap in the latent space. Though Move-AE was trained to compress its inputs by a much larger ratio (420: 1 for Move-AE vs 7: 1 for Pose-AE and PCA), it was able to embed whole movements in different regions of the latent space without explicit labels. We observed many variations in how the data were embedded in the latent space between each of the 10 runs, especially for Move-AE, hence the increased variability in classification performance.

Movement Classification

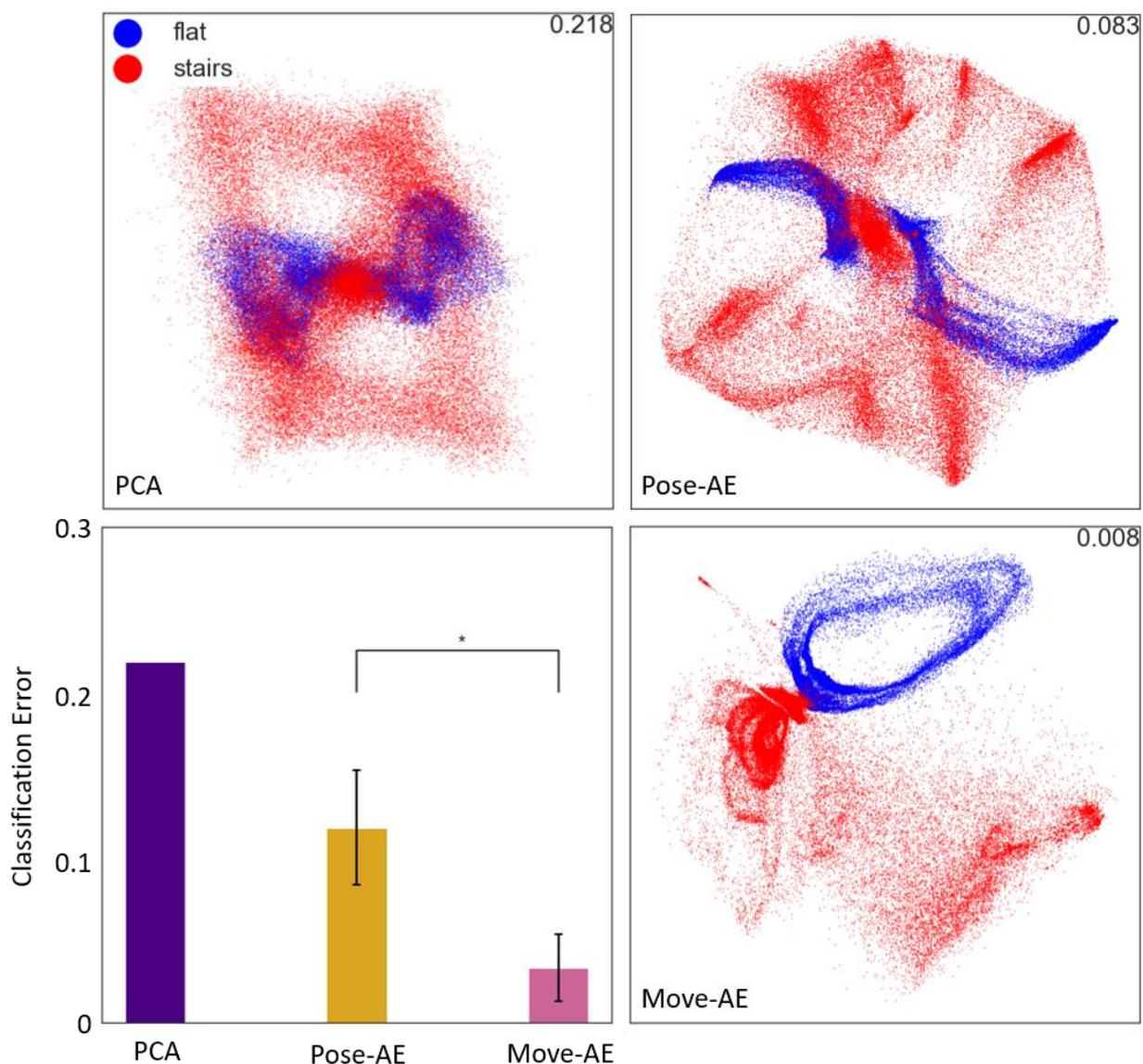


Figure 4-4. The three panes display the best performing latent space of their respective method. The latent space is a visualization of the activations of the two coding units in the bottleneck, or the first two principal components. The bottom-left pane shows the results of a movement classifier SVM trained on the latent spaces for each method. Error bars denote the varying performance from each of 10 runs for both autoencoders. The Move-AE latent space outperformed Pose-AE and PCA latent spaces on classifying between flat ground and stair walking. Recall that in the Move-AE

latent space, each point represents an entire movement. Flat ground walking is embedded in a cyclical structure that is well separated from stair walking.

Individual Identification

Similar to movement classification, Move-AE outperforms Pose-AE (KS test, $p < 0.0001$), which in turn outperforms PCA on classification of individuals (error 62.0% PCA; $48.9 \pm 2.6\%$ Pose-AE; $28.9 \pm 9.3\%$ Move-AE). All three methods produce cyclical representations of gait within their latent spaces, but Move-AE also cleanly separates between many individual gaits, again without providing an explicit label (*Figure 4-5*).

Individual Identification

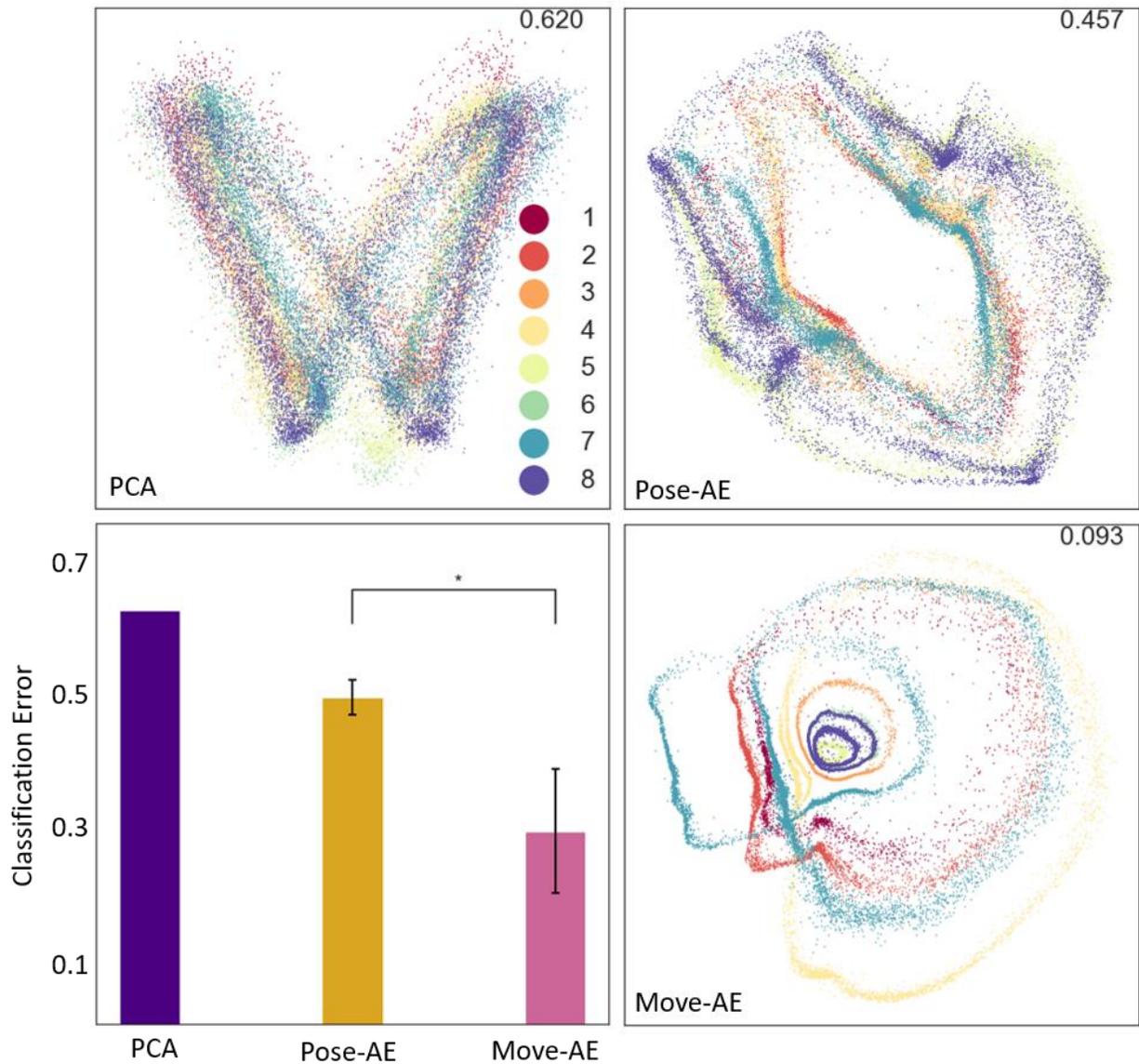


Figure 4-5. The three panes display the best performing latent space of their respective method. The plotted data is colored by one of eight individual subjects. The classification error for the best performing model is included in the top-right of each plot. The bottom-left pane shows the results of an individual-specific classifier SVM trained on the latent spaces for each method. Note that the Move-AE latent space here is a well-performing outlier, though visual inspection of the other latent spaces confirms the general behavior of separately embedding individuals.

Discussion

Understanding the high dimensionality of human gait remains a significant challenge but may yield an equally significant payoff. Creating a useful low dimensional representation of gait may serve to benefit both gait analysis and control of a device. Many techniques for dimensionality reduction exist, though PCA has remained popular for its ease of implementation and interpretability. However, our results indicate that autoencoders are better suited for reducing human movement on both performance measures of reconstruction and Variance Accounted For. Relationships with human gait features, and biological data in general, is generally nonlinear. The nonlinear activation function within neural networks enables them to capture such relationships, whereas PCA cannot. Nonlinear PCA methods like Kernel PCA may be better suited than standard PCA for such tasks (Mika et al., 1998).

We employed three datasets of increasing complexity. Flat ground walking is the least complex, in that it only contains cyclic steady-state gait. Stair walking is more complex, as it contains upstairs and downstairs segments, and the transitions in between. Natural movements are most complex, in that they contain both cyclic movement and non-cyclic movements, with a variety of actions being performed. As data becomes more complex, the advantage of autoencoders over PCA is diminished (*Figure 4-2*). This may be due to the tradeoff between quantity and variety of movements within each dataset, thus the autoencoder is impoverished of sufficient examples of more complex movements from which to learn. Thus, care should be taken to carefully curate the activities within smaller datasets to achieve good dimensionality reduction. For instance, composition of the training data should be deliberately balanced to match the desired performance on each example. Movements that appear rarely will not affect the gradient sufficiently to achieve

adequate reconstruction, whereas movements that appear too often will dominate the gradient at the expense of others.

Incorporating temporal context in the input to Move-AE dramatically enhances its capability to discriminate between movements and individual gait profiles. This is understandable considering how given a full second of gait, or about one full gait cycle, whatever differences that exist between individuals or movements will be present within every input. This capability is not afforded to standard PCA, which can only operate on n -dimensional vectors, rather than $m \times n$ -dimensional matrices. Interestingly, producing Move-AE also had some unintended consequences. For instance, while performing the individual identification task, it became apparent that half of subject 7's flat ground data was persistently embedded separate from all others in the latent space. Upon visual inspection of the data, it was apparent there was a minor sensor calibration or data processing error that was small enough to escape detection until that moment (offending data was removed and all trials repeated without it).

We show that it is possible to reliably classify movements and some individuals using the Move-AE architecture. However, without retraining the network on an equivalent dataset of actual prosthesis users, it is unknown how effective such a strategy may be in practice. Nevertheless, automating the selection of modes, or perhaps gait parameters, reduces the control burden on the users of mode-based prostheses, who must perform unnatural motions with their prosthesis to select the right mode for the terrain. We also show that Move-AE is sensitive to individual gaits. These variations, arising from the dynamic cost landscape of walking, are important to preserve. This also portends that such a network will be able to capture the dynamics of pathological or compensatory gait, embedding them within discrete latent structures. However, organization of

individuals in the latent space is not necessarily meaningful - sampling halfway between two individuals in the latent space may not produce a pose or movement that is halfway between them in Euclidean space.

It should be noted that the aim of the classification task presented here is not intended to maximize classification performance of movements. There are other, better suited methods to achieve high classification accuracy, when labelled data is available. Rather, we designed the task to compare the relative capacity of each dimensionality reduction method to preserve valuable information like movement class or individual gait. Future work is needed to determine how such methods perform on unseen movements and individuals. Autoencoders in particular tend to “de-noise” unseen data such that it better resembles the data on which they were trained (Vincent et al., 2008).

Although dimensionality reduction techniques as described here are powerful tools to simplify and analyze gait data, they are not sufficient to achieve prosthetic control alone. An autoencoder on movement data only serves to make sensor data more palatable—it does not provide its own inference about the data. For instance, the Move-AE architecture self-supervises to embed movements in the latent space, but does not classify without an additional classifier like an SVM. Our results show that such learned embeddings can automatically separate movements in the absence of any goal but reconstruction. Future work is needed to move the application of these tools from offline analysis to online integration with a controller. In a practical, online scenario, it is still unknown the quantity, variety, and richness of data required from an individual walker to train a personalized Move-AE architecture to satisfaction towards a given task, like movement classification. Indeed, this discussion focuses on facilitation of “high-level” prosthetic control, or mode selection, rather than “low-level” control over moment-to-moment commands to the actuator

which remains critically important. Furthermore, our results may still be replicable using raw inertial measurement unit data from a few strategically placed sensors on the lower limbs, rather than the full wearable motion capture system used here, in line with the work by (Geissinger and Asbeck, 2020).

We have demonstrated that autoencoders can generate structured, interpretable latent spaces. This class of self-supervised networks are able to learn without hand-crafted labels, making them suitable to tackle complex problems like human movement. For instance, latent representations of gait form cyclic structures organized by phase, without human intervention to segment the gait data. Though not presented here, the authors found that distance to the center of the cyclic structure corresponds directly with cadence—faster cadences form tighter rings, slower cadences form larger ones. Contrary to the popular notion that neural networks are a black box, autoencoders can produce structured latent spaces, and thus could be incorporated into prosthetic controllers, either to simplify incoming sensor data, or to generate movement commands via sampling in the latent space.

Interpretation of latent spaces is fast becoming an important topic of research as neural networks become more prevalent. It should be noted that sampling from these latent spaces may enable generation of individual-specific synthetic gait cycles. For instance, sampling points from a Gaussian distribution centered on the region where mid-swing is embedded in the latent space may produce multiple variations of a mid-swing trajectory in the decoder, as learned from training data. Further research is needed to determine how best to create a latent space that lends itself to sampling - as stated previously, distances within an autoencoder's latent space are not necessarily meaningful. Sampling meaningful movements from the latent space is a non-trivial problem, in

part due to the difficulty in describing the latent space's geometry, or manifold. Application of adversarial or variational autoencoders, which enforce additional distributional constraints on the latent space, may be key to building sample-suitable latent spaces.

Useful representations of gait are a necessary ingredient for leveraging the power of machine learning for prosthetic control. This study shows how autoencoders may create such a representation purely from data, and crucially, are capable of handling temporal data.

Chapter 5 Learning to Operate a High-Dimensional Hand via a Low-Dimensional Controller

Introduction

The complexity of a human hand has been the topic of numerous research aimed at understanding its underlying control strategies. With 27 degrees of freedom (DOFs) controlled by 34 muscles, replacement of the hand, in cases of congenital or acquired amputation, can be a difficult task, oftentimes either oversimplified (e.g., one-dimensional hooks) or overcomplicated (e.g., high-dimensional prosthetic hands) by prosthetic solutions. And while the intricacy of developed prosthetic hands available on the market grew over the last five decades (Belter et al., 2013), their control methods have fallen behind (Castellini, 2020).

The conventional method of controlling dexterous prosthetic hands is through myoelectric interfaces, in which electromyographic (EMG) signals from existing muscles in the amputee's residual limb are used to operate the device. However, lack of available clean muscle signals due to the difference in amputation levels oftentimes poses limitations on the controllers themselves (O'Neill et al., 1994). The issue arises from the fact that while there might be many DOFs in the device, which allows for an individuated movement, a limited number of EMG signals might be available on the residual limb to control these DOFs (Iqbal and Subramaniam, 2018). To account for the differences in the control and output dimensions, some have investigated the potential of using dimensionality-reduction (DR) methods.

One of the most famous of all studies in which a DR technique was applied to complex hand kinematics during object grasping was the study by Santello et al. (Santello et al., 1998). There, the group used principal component analysis (PCA), a linear DR technique that creates a low

dimensional (latent) representation of the data by finding the directions in the original space that explain the most variance in the input data. In their study, they found that a two-dimensional latent space could account for approximately 80% of the variability of hand kinematics during grasping. Relying on this finding, several groups have developed, what they called, a *postural controller* in which a prosthetic hand with multiple DOFs could be operated through a 2D space (Magenes et al., 2008; Ciocarlie and Allen, 2009; Matrone et al., 2010; Matrone et al., 2012; Segil, 2013; Segil and Controzzi, 2014; Segil, 2015).

One of the main limitations of PCA is its linearity, due to which it can only account for linear relationship in the input data. In our recent study, we explored the use of a nonlinear autoencoders (AEs) as a way to account for nonlinear relationships in hand kinematics data (Portnova-Fahreeva et al., 2020). In the study, we found that two latent dimensions of an AE could produce superior results to that of its linear counterpart, PCA, reconstructing over 90% of hand kinematics data. In addition, a nonlinear AE allowed to spread the variance more uniformly across its latent dimensions, which can consequently allow for a more uniform distribution of control across each DOF.

With such superior characteristics, AEs may serve as a platform for more accurate lower-dimensional prosthetic control, utilizing its reconstruction power and more equal spread of latent dimension variance. Based on these findings, our group developed a myoelectric interface that allowed for the low-dimensional (2D) control of a hand with 17 DOFs that is described further in this paper. In this paper, we refer to it as *AE-based controller*.

But is the dimensionality mismatch between device DOFs and control signals the only issue when it comes to myoelectric prosthetic control? Or is the problem at hand (figuratively and actually) more complex?

We attempted to answer these questions with three studies where we trained the participants to perform various hand gestures on a virtual hand with 17 DOFs via a low-dimensional (2D) controller (*Figure 5-1*). The studies assessed the factors that contribute to the difficulty of a myoelectric hand control and design learning paradigms that address those factors. The explored factors were the difficulty of novel myoelectric interfaces and implicit and explicit training of the underlying control dimensionality.

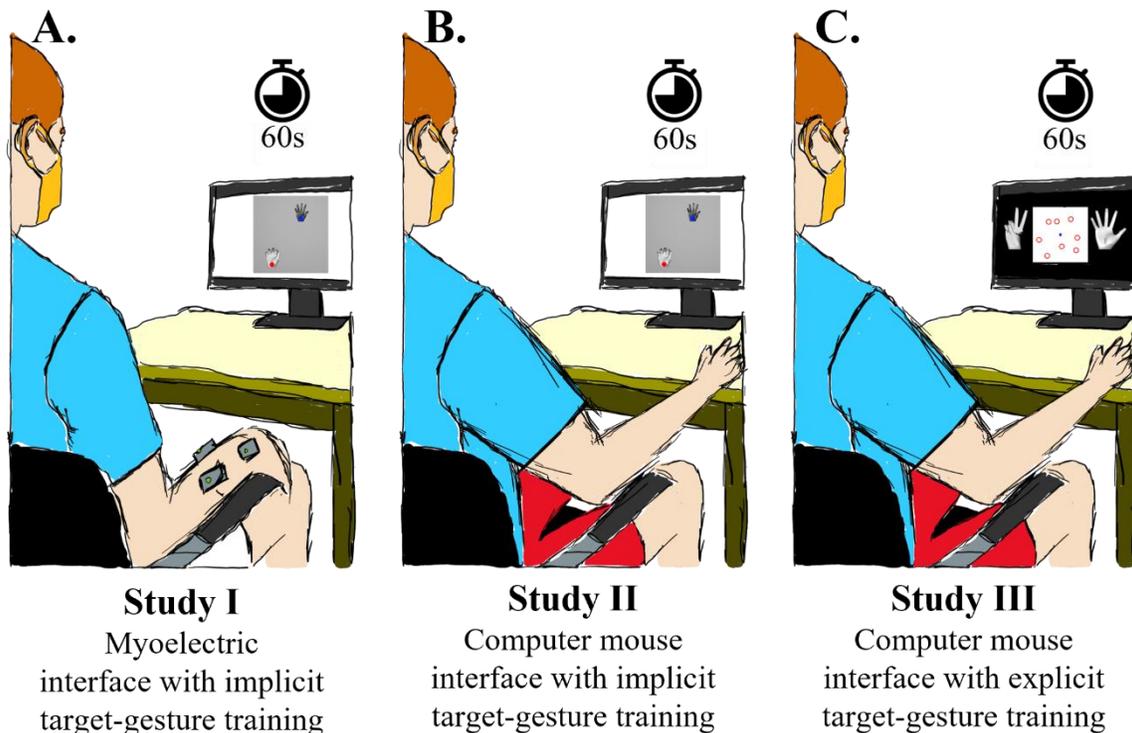


Figure 5-1. Experimental setup. (A) Study I had a long exploration window of 60s and implicit target-gesture training via a myoelectric interface. (B) Study II had a long exploration window of

60s and implicit target-gesture training via a mouse-based interface. (C) Study III incorporated explicit target-gesture training with a long exploration window of 60s and a mouse-based interface.

Study I employed a myoelectric interface in which the participants controlled a virtual hand via a low-dimensional 2D controller with the use of their wrist muscles: flexion/extension and abduction/adduction.

Study II employed a computer mouse to assess how much of the difficulty of learning the controller was from the difficulty of understanding to operate an EMG-based interface.

Additionally, Studies I and II explored how having an additional session of target reaching in 2D affected the performance of controlling a virtual hand for both myoelectric and mouse-based interfaces.

Lastly, Studies II and III explored how much of the difficulty in the learning arose from the user's inability to establish the connection between the underlying 2D control and the actual 17D virtual hand. Like Study II, Study III employed a mouse-based interface, but included a modified target-reaching session in 2D to establish an explicit connection for the user between the dimensionalities of the underlying control and the presented task.

Methods

Nonlinear Postural Controller

Using the findings of our previous study (Portnova-Fahreeva et al., 2020), we developed a low-dimensional controller, in which a 17-DOF virtual hand was operated using two control signals. In that study, we looked at the performance of an AE structure (*Figure 5-2A*) and determined its

superiority to the conventional linear PCA method when reducing the dimensionality of complex hand kinematics to two latent dimensions.

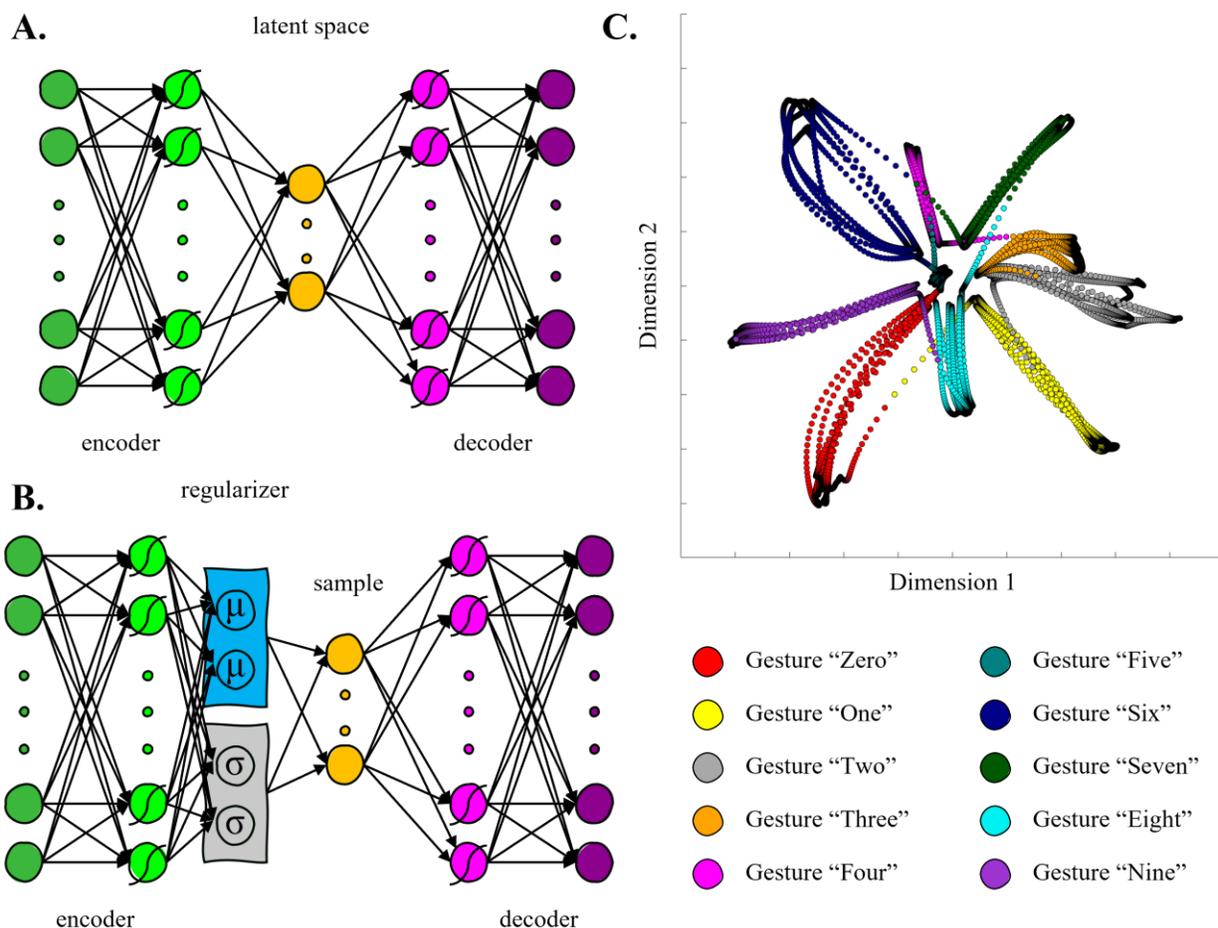


Figure 5-2. (A) Structure of a regular autoencoder (AE) with three hidden layers. The middle layer represents the latent space. Curves over the neurons represent layers obtained with a nonlinear activation function. (B) Structure of a variational autoencoder (VAE) with three hidden layers and a regularizer term before the latent space. (C) Latent space derived by applying a variational autoencoder to hand kinematics data of an individual performing American Sign Language (ASL) gestures.

AEs are artificial neural networks consisting of two components: an encoder that converts the inputs (x) to a latent representation, followed by a decoder that transforms the latent representation into the outputs (\hat{x}), with the same dimensions as the inputs. They learn to efficiently encode the

input data variability within its latent space by minimizing the reconstruction error between the output and the input of the network (*Equation 1*).

$$Loss_{AE} = \|x - \hat{x}\|^2 \quad (1)$$

Variational Autoencoder

Despite their strong capabilities of reconstruction of biological data, standard AEs have a fundamental problem of creating a latent space whose topological characteristics inhibit easy and intuitive interpolation. Points that are not encoded onto the latent space often reconstruct to unrealistic data. In the case of the hand kinematic data from our previous study (Portnova-Fahreeva et al., 2020), this would result in the reconstruction of impossible gestures with joint angles outside of their possible ranges of motion. As a result, a latent space like this may not be the most optimal option for myoelectric prosthetic control.

To counteract the fundamental problem of standard AEs for the development of our controller, we propose using a Variational Autoencoder (VAE) (Kingma and Welling, 2013). Differently from a standard AE, a regularizer term is added to the reconstruction error in the VAE cost function and aims to match the probability distributions of the latent space to that of a given distribution set prior to model training (*Figure 5-2B*). VAEs typically use the Kullback-Leibler Divergence (KLD) (Kullback and Leibler, 1951) to minimize the distance between the latent and the source distribution. The cost function that VAE optimizes is shown in *Equation 2*.

$$Loss_{VAE} = \|x - \hat{x}\|^2 + \beta * KLD[N(\mu_x, \sigma_x), N(0, I)] \quad (2)$$

By optimizing the two terms of the cost function, the resulted VAE latent space can locally maintain the similarity of nearby encodings via clustering yet is globally densely packed near the

latent space origin. This VAE structure may have an advantage of creating a usable 2D latent space for a postural control. In our study, the VAE was trained to regularize the latent space distribution $N(\mu_x, \sigma_x)$ into a normal Gaussian distribution ($\mu = 0, \sigma = 1$). If achieved, the space would exhibit a shape desired for center-out target-reaching tasks that we designed for the three studies.

We performed hyperparameter tuning on the VAE network to determine the most optimal model for the given data. We used a separate validation dataset not included in the analysis performed in the original study, in which the participant performed American Sign Language (ASL) gestures. The hyperparameters under assessment were number of training epochs, nonlinear activation function between neural network layers, learning rate, and the weight on the regularizer term in the cost function, indicated as β (Equation 2). The performance of each hyperparameter pair was evaluated in terms of reconstruction, assessed with a root-mean-square error (RMSE) between the input and the output of the network, and closeness of the latent space distribution to that of the normal Gaussian, calculated with the KLD.

After hyperparameter tuning, the VAE network was applied to the data recorded from one the participants (*PI*) from Portnova-Fahreeva et al. and resulted in a 2D latent space with separable gesture classes and a shape that we aimed for the purpose of the center-out target-reaching task in the following studies (Figure 5-2C).

Kinematic Decoding

The resulted 2D latent space was consequently used for Studies I, II, and III to represent the lower-dimensional control space. The decoder sub-network of the VAE model was utilized to reconstruct a point on a 2D control space (x_{latent}, y_{latent}) to 17D virtual hand kinematics (J_{hand}), where w_i

and b_i were weights and biases of the VAE decoder network layers (*Equations 3, 4*). Note that the decoder layers are the fourth and fifth of the overall VAE network (*Figure 5-2B*).

$$layer_4 = \tanh ([x_{latent} \quad y_{latent}] * w_4 + b_4) \quad (3)$$

$$J_{hand} = layer_4 * w_5 + b_5 \quad (4)$$

With this controller, a user was able to control a high-dimensional virtual hand by moving a point on a 2D plane. Different modalities of control input were applied in Studies I, II, and III – in Studies I, the participants moved the 2D controller via a myoelectric interface whereas the participants used a computer mouse interface in Studies II and III.

Virtual Hand

For all studies, we developed a virtual environment, in which we used a 3D computer model of a hand with 17 DOFs (*Figure 5-3A*). The 17 DOFs that were controlled were flexions/extensions of the three joints (metacarpal, proximal interphalangeal, distal interphalangeal) of the four fingers (pinky, ring, middle, and index) and flexion/extension of two joints of the thumb (metacarpal and interphalangeal) as well as the 3D rotation of its carpometacarpal joint.

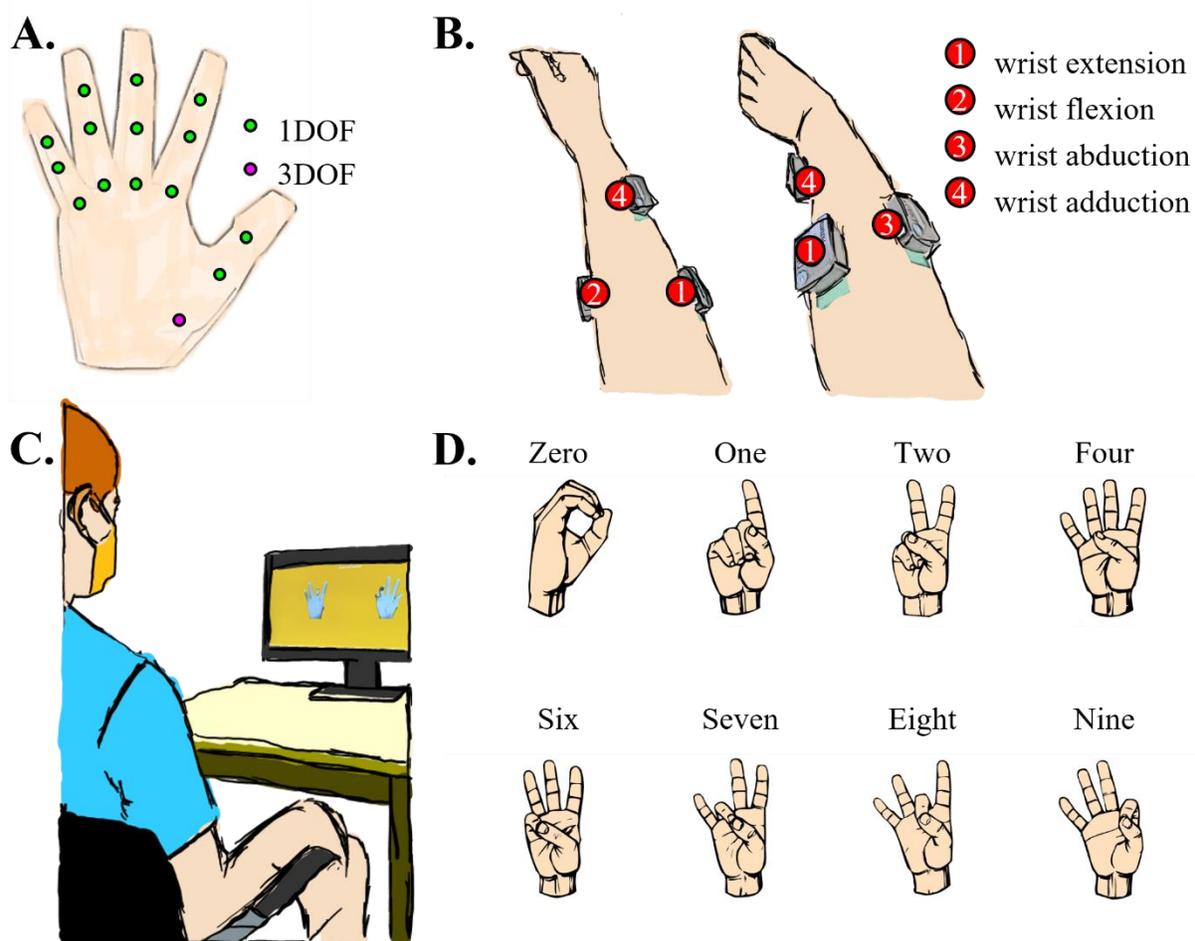


Figure 5-3. (A) 17 degrees of a freedom (DOFs) of the virtual hand. (B) Surface electrode placement on the participant's forearm in the myoelectric interface. Each electrode location is specific to one of the four wrist movements. (C) Experiment setup with the participant sitting approximately 1.5m away from the computer screen during one of the studies. (D) Eight American Sign Language (ASL) gestures that the participants were being trained to recreate during the studies.

The plane that the users were operating on during the controller was defined by a square that enclosed all the points on the hand kinematics latent space defined in *Figure 5-1C*. As a result of that, locations on the 2D plane that were not defined as part of the encoded points could result in reconstructed postures of impossible kinematics. To prevent the hand from generating biologically unnatural gestures during the control, possible ranges of motion of the virtual hand joints were limited to the ranges of motions of an actual hand.

Myoelectric Interface

In the myoelectric interface, the control was performed with muscle signals acquired from four surface EMG electrodes (*Delsys Inc, MA, USA*), placed on the participant's right forearm at the location of four muscles for wrist movement (*Figure 5-3B*). Proper electrode placement for myoelectric control was verified in a separate calibration task described in the *EMG Calibration* section below.

Raw EMG signals were recorded at a sampling frequency of *2kHz*. A series of standard pre-processing techniques were applied to the raw recordings to extract the EMG envelope: band pass filter *30 – 450Hz* to eliminate movement artifacts and high frequency noise, rectification, and low-pass filter at *3Hz*.

EMG envelopes were mapped into hand kinematics as follows: vector summation algorithm of EMG, EMG-to-kinematics map, and hand kinematics decoding (*Figure 5-4*). We described the hand kinematics decoding in the earlier section of *Kinematic Decoding*. Vector summation and EMG-to-kinematics are described in the *Vector Summation Algorithm of EMG* and *EMG-to-Kinematics Mapping* sections, respectively. Before that, we outline a calibration procedure for the recorded EMG signals that ensured participants had full coverage of the 2D control space.

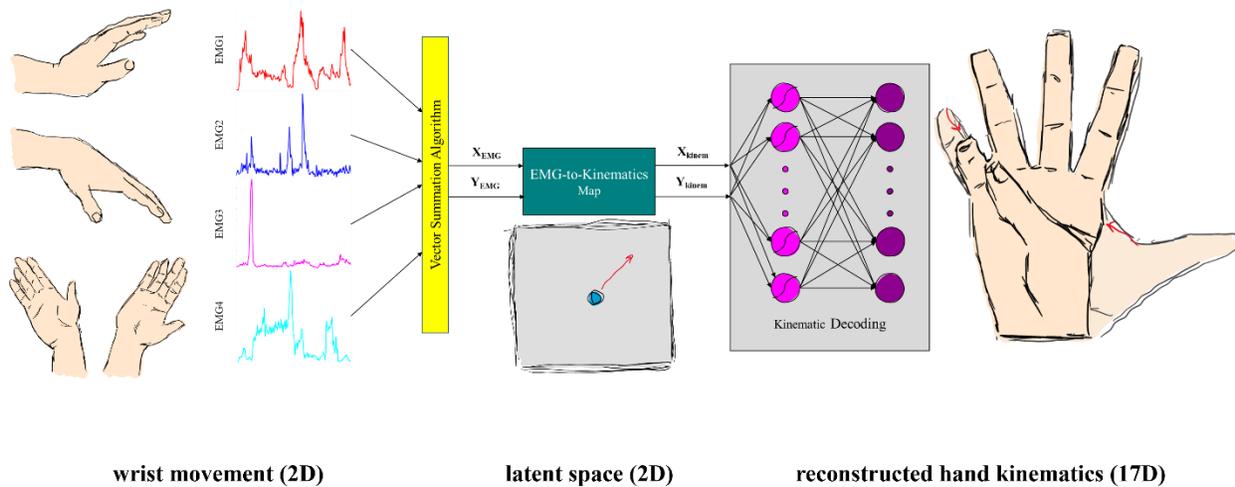


Figure 5-4. Setup of a myoelectric interface. Wrist movements generated EMG signals, which, in turn, were combined using a Vector Summation Algorithm into a 2D vector (X_{EMG}, Y_{EMG}). The vector, in turn, controlled the position of a 2D cursor on the latent space. Every point on the latent space (X_{kinem}, Y_{kinem}) reconstructed into full 17D hand kinematics via the decoder part of the variational autoencoder network.

EMG Calibration

Prior to Study I, each participant was asked to perform 60s of structured movements, in which they were asked to move their wrist up/down/right/left from the neutral position (hand resting upright on the lap). Each movement was performed seven times with the guidance of the study coordinator. The participants were asked to keep contractions at a comfortable level when performing the movements. They were also advised to perform each movement activating muscles as independently of the others as possible.

In addition, we recorded 10s of resting EMG, in which the participants had their right hand placed on their lap in a comfortable position, with the muscles completely relaxed. The signals recorded during this resting phase were used to subtract from each muscle's offsets that did not correspond to voluntary contractions of the muscles.

For each muscle i , the EMG signal was calibrated using the maximum value of the envelope recorded during rest, $\max(EMG_{rest,i})$, and during the structured movements, $\max(EMG_{struct,i})$ (Pistohl et al., 2013) (Equation 5). A scaling value, $scale_i$, was also computed to ensure the participants had full coverage of the workspace without over-contracting their muscles.

$$EMG_{calib,i} = scale_i \frac{EMG_i - \max(EMG_{rest,i})}{\max(EMG_{struct,i}) - \max(EMG_{rest,i})} \quad (5)$$

Vector Summation Algorithm of EMG

The calibrated EMG signals of the four wrist muscles were combined using a vector summation algorithm (VSA) to obtain a 2D control signal. The muscles that controlled wrist extension/flexion were mapped to move the 2D controller in the up/down direction (*i.e.*, y axis). Similarly, wrist abduction/adduction moved the controller in the right/left direction (*i.e.*, x axis). An offset was also added to both the x and y directions in cases when the calibrated rest position did not appear to match the center point of the workspace (Equations 6, 7).

$$x_{cursor} = (EMG_{abd,calib} - EMG_{add,calib}) - x_{offset} \quad (6)$$

$$y_{cursor} = (EMG_{ext,calib} - EMG_{flex,calib}) - y_{offset} \quad (7)$$

Matching the resting EMG position with the center of the latent space during calibration ensured that every trial started from the neutral position and every movement was performed in the center-out reaching manner. In the resting position, the corresponding virtual hand position was with all five fingers completely open (*neutral gesture*).

An additional low-pass filter of 1Hz was applied to the cursor position to reduce the signal-to-noise ratio and stabilize the myoelectric controller.

EMG-to-Kinematics Mapping

After mapping calibrated EMG signals to a 2D cursor position on the screen (x_{cursor}, y_{cursor}), we then transformed into a 2D position on the latent space (x_{latent}, y_{latent}). It was done by scaling the cursor position using $screen_{max}$, length of control plane in the local coordinate frame on the screen, and $latent_{max}$, length of control plane in the latent space (Equations 8, 9). This was done to account for the difference in the screen and latent space dimensions.

$$x_{latent} = x_{cursor} * \frac{latent_{max}}{screen_{max}} \quad (8)$$

$$y_{latent} = y_{cursor} * \frac{latent_{max}}{screen_{max}} \quad (9)$$

Mouse Interface

In the mouse interface, the participants could operate the virtual hand by clicking and holding the left button of their computer mouse. Dragging the mouse across the screen, in turn, controlled the position of the 2D controller. Like the myoelectric interface, the mouse-based interface employed the same transformation between the screen and latent space reference frames (Equations 8, 9).

White Gaussian noise ($\mu = 0, \sigma = 0.02$) was added to the position of the controller to recreate the intrinsic neuromuscular noise often found in myoelectric interfaces due to the nature of EMG acquisition. An additional low-pass filter of 1Hz was applied to the cursor position to recreate the pre-processing delay experienced in the previous myoelectric interface.

Releasing the mouse button returned the control position back to the center of the 2D plane, which, in turn, corresponded to the neutral gesture in the virtual hand (all five fingers completely open).

Overall Protocol

No physical constraints were imposed on the participants throughout the experiment as they were free to move their right arm while performing the experiment objectives. The only constraints were placed during data collection for EMG calibration purposes. More details can be found in the *EMG Calibration* section.

During each study, the participants were seated in an upright position in front of a computer screen, at approximately 1.5m away at eye level (*Figure 5-3C*). Over the span of one-two hours, they engaged in different training tasks to learn to recreate gestures in a virtual 17D hand via the myoelectric or mouse interface. The gestures that they were required to recreate were eight ASL gestures, representing the numbers “zero”, “one”, “two”, “four”, “six”, “seven”, “eight”, and “nine” (*Figure 5-3D*). The training tasks differed based on the assigned group of each participant. There was a total of three groups:

Absence of 2D training (17D Group)

For the 17D group, the dimensionality of the visual feedback of the task did not match that of the underlying control interface. This means that the participants were always presented with two virtual hands – the hand to match (*left, Figure 5-5A*) and the hand to control (*right, Figure 5-5A*) – without any presentation of the underlying 2D control space. We referred to this as a *17D task* (*Figure 5-5A*). More detailed information on the structure of the tasks can be found in the *Study I* section below.

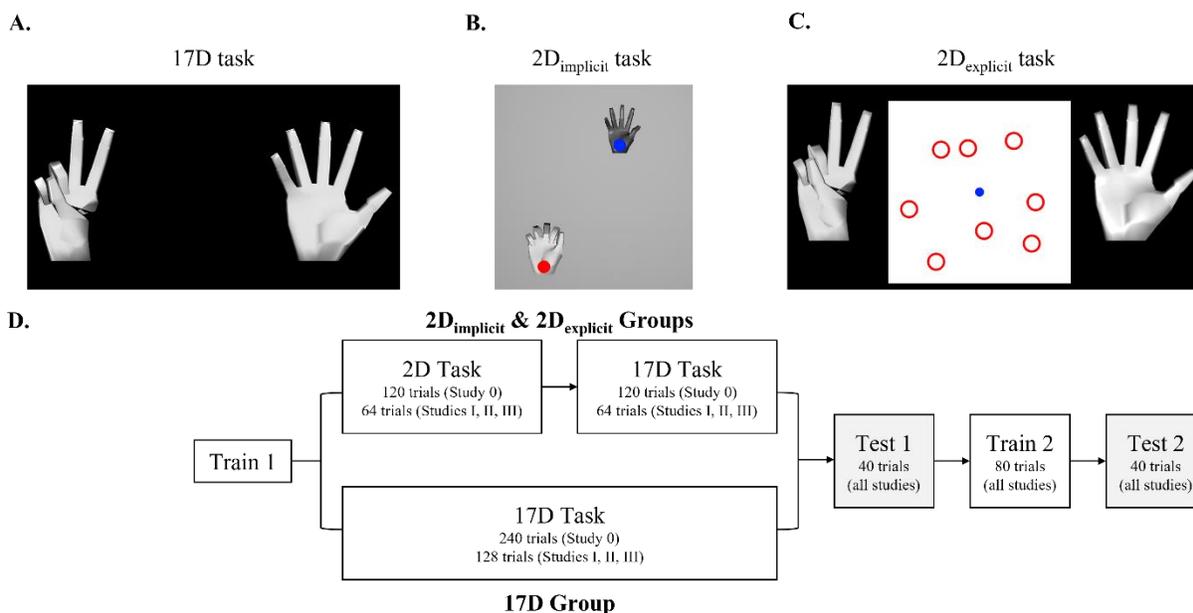


Figure 5-5. (A) Setup of the 17D task with two virtual 17D hands present. The hand on the left is the target hand that the participants needed to match. The hand on the right is the controlled hand that the participants controlled either via a myoelectric or a mouse interface. (B) Setup of the $2D_{\text{implicit}}$ task where the participants performed simple 2D reaches – with the blue controlled cursor and the overlaid hand to the red target and its corresponding hand. (C) Setup of the $2D_{\text{explicit}}$ task, in which the participants controlled the blue cursor on the 2D plane, which in turn controlled the 17D of the hand on the right. The hand on the left was the target hand. The participants were required to learn which of the eight present red targets represent the 2D location of the target gesture. (D) Sequence of training and test session in each study. The only difference between the groups is in *Train1* session, where the 17D group only experiences the 17D task while the $2D_{\text{implicit/explicit}}$ groups have the session split in half: $2D_{\text{implicit/explicit}}$ task and 17D task.

Implicit 2D Training ($2D_{\text{implicit}}$ Group)

For this group, the dimensionality of the visual feedback of a training task matched that of the underlying control interface. Prior to being exposed to the 17D task, this group had an initial training where they could practice a series of target reaches in 2D, referred to as a $2D_{\text{implicit}}$ task. The virtual hands that they matched and controlled were presented over the 2D target and the cursor, respectively (*Figure 5-5B*). Targets were presented one at a time. Both hands changed their kinematics in accordance with their location on the 2D plane. However, *no explicit* explanation

was given to the participants on how the target-reaching task related to the 17D task they were later presented with.

Explicit 2D Training ($2D_{\text{explicit}}$ Group)

Similar to the $2D_{\text{implicit}}$ group, the $2D_{\text{explicit}}$ group had an initial training on the 2D control space prior to the 17D task. We referred to this as a $2D_{\text{explicit}}$ task. There, the participants were presented with eight red targets, which, in turn, represented 2D position of the eight ASL gestures on the latent space (*Figure 5-5C*). They were given visual feedback of the target (*left, Figure 5-5C*) and the controlled (*right, Figure 5-5C*) hands alongside the control plane rather than overlaid on the cursor position as in the $2D_{\text{implicit}}$ task. Both hands changed their kinematics in accordance with their location on the 2D plane. In addition, the participants in this group were specifically instructed that cursor movements were directly related to the kinematics of the presented hand and, therefore, were explicitly instructed on the connection between the underlying dimensionality of the controller and the generated hand gestures.

Each study was divided into four blocks: *Train1*, *Test1*, *Train2*, and *Test2*. *Train1* session was split in two different parts of equal lengths for the $2D_{\text{implicit}}$ and $2D_{\text{explicit}}$ groups: $2D_{\text{implicit}}/2D_{\text{explicit}}$ task and 17D task (*Figure 5-5D*).

Next, we outline the protocol details for each of the three studies.

Study I

The goal of Study I was 1) to verify the ability of the users to learn how to operate the lower-dimensional controller through a myoelectric interface and 2) to understand the effect of varying the dimensionality of visual feedback during training on learning the complex 17D gesture-

matching task with an underlying 2D control (*Figure 5-1A*). It was designed after classical center-out target-reaching studies, in which the participants were given a short exploration window to successfully complete a single trial (Pistohl et al., 2013; Segil, 2013; 2015; Dyson et al., 2018).

For this study, we recruited 14 unimpaired right-handed individuals (four males, ten females, 25.6 ± 5.9 years old). All participants were naïve to the myoelectric interface. Participant recruitment and data collection conformed with the University of Washington's Institution Review Board (IRB). Informed written consent was obtained from each participant prior to the experiment.

The participants were randomly split into two groups based on different learning paradigms (17D and 2D_{implicit}).

Training Sessions

The main difference between the two participant groups was in *Train1*. The 2D_{implicit} group practiced the 2D_{implicit} task for the first half of *Train1* and switched to the 17D task for the second half of the session (*Figure 5-5D*). The 17D group performed the 17D task for the entirety of the session. More information on the differences between the tasks can be found below.

2D_{implicit} Task

During the 2D_{implicit} task, the participants engaged in a center-out target-reaching task. They were presented with visual feedback of the cursor that they controlled and different targets that they needed to reach (*Figure 5-5B*). The targets and the cursor were represented with circles of the same size (approximately 0.25" radius). Target locations were placed at various distances from the center cursor, which effectively were the locations of the eight ASL gestures on the latent space.

Grey and white hands were placed over both the control cursor and the target, respectively. Both hands showed the reconstructed gestures related to the current position of the cursor and the target on the latent space using the VAE decoder described in the *Kinematic Decoding* section. The grey hand was slightly smaller in size than the white one for ease of differentiation once the two hands overlaid each other.

The participants were given 60s to reach the targets. If the cursor was within the acceptable range from the target, both the hands and the target turned yellow for visual feedback. The acceptable range was equivalent to 0.5 units from the center of the target on the latent space (or 0.25" on the screen). The target was successfully reached if the cursor stayed within the acceptable range for 0.75s. Upon successful completion of each trial, the hands and the target turned green.

At the end of each trial, successful or not, the participants heard a sound cue (“Relax your muscles”), asking them to relax their muscles, which, in turn, returned the cursor back to the neutral position in the center of the control plane. Once the participant was completely relaxed for 1.5s, a new target was presented, and a sound cue (“Go”) was given to start the next reaching trial.

During the 2D_{implicit} task of *Train1*, ASL gestures were presented for a total of 64 trials (*i.e.*, eight gestures repeated eight times) in a pseudo-random order. Gestures of the same type were never presented consequently one after another. The participants were given one minute to rest after every 32 trials.

17D Task

During the 17D task, the participants were presented with only two virtual hands (*Figure 5-5A*) and had no visual feedback about the location of the cursor and the target on the 2D latent space.

The hand on the left was the hand the participants needed to match. The hand on the right was the hand controlled with the wrist muscles through the myoelectric interface. A sound cue indicated the beginning of a new trial. Each trial always started from a neutral pose, which corresponded to the cursor placement in the center of the latent space and was generated by completely relaxing the muscles.

At the beginning of a new trial, the matching hand would form a new gesture and the participants had 60s to match and hold it with the controlled hand within its acceptable range. The acceptable range in the 17D task was determined by the 2D control space (*i. e.*, if the 2D cursor related to the current hand gesture was close enough to the 2D target representing the gesture of the matching hand, then the controlled hand was within the acceptable range). Note that the acceptable range condition was the same as that of the 2D_{implicit} task. The participants were required to maintain the gesture for 0.75s for the trial to be counted as successful.

During the 17D task of *Train1*, the 2D_{implicit} group got trained for a total of 64 trials (*i.e.*, eight gestures repeated eight time) in a pseudo-random order. The 17D group experienced the 17D task for 128 trials total (*i.e.*, eight ASL gestures repeated 16 times) during *Train1*. The participants in both groups were given one minute to rest after every 32 trials.

Sound cues at the beginning (“Go”) and end (“Relax your muscles”) of each trial during the 17D task were the same as they were during the 2D_{implicit} task.

During *Train2*, both groups performed the 17D task, where the eight ASL gestures were repeated 10 times in a pseudorandom order, for a total of 80 trials per session. One minute break was given

to the participants after 40 trials. The time-to-reach and time-to-hold were the same as in *Train1* with the same sound cues at the beginning and end of each trial.

Test Sessions

The goal of the test sessions was to determine whether the participants were able to transfer the skills acquired during training to conditions where they needed to recreate slightly different gestures. During both *Test1* and *Test2*, the participants were asked to match the hand gesture that they saw on the right, similar to the 17D task during training.

After hearing a sound cue, they had 6s to successfully match the gesture on the left with the hand on the right, with a 0.75s of holding time within the acceptable range. The gestures that they were required to match during test sessions were slight modifications of the gestures they got trained on. They were created by reconstructing a point that was 75% along the path to the gesture on the latent space (*Figure 5-6*). Each new trial started and ended with the same sound cues as during training.

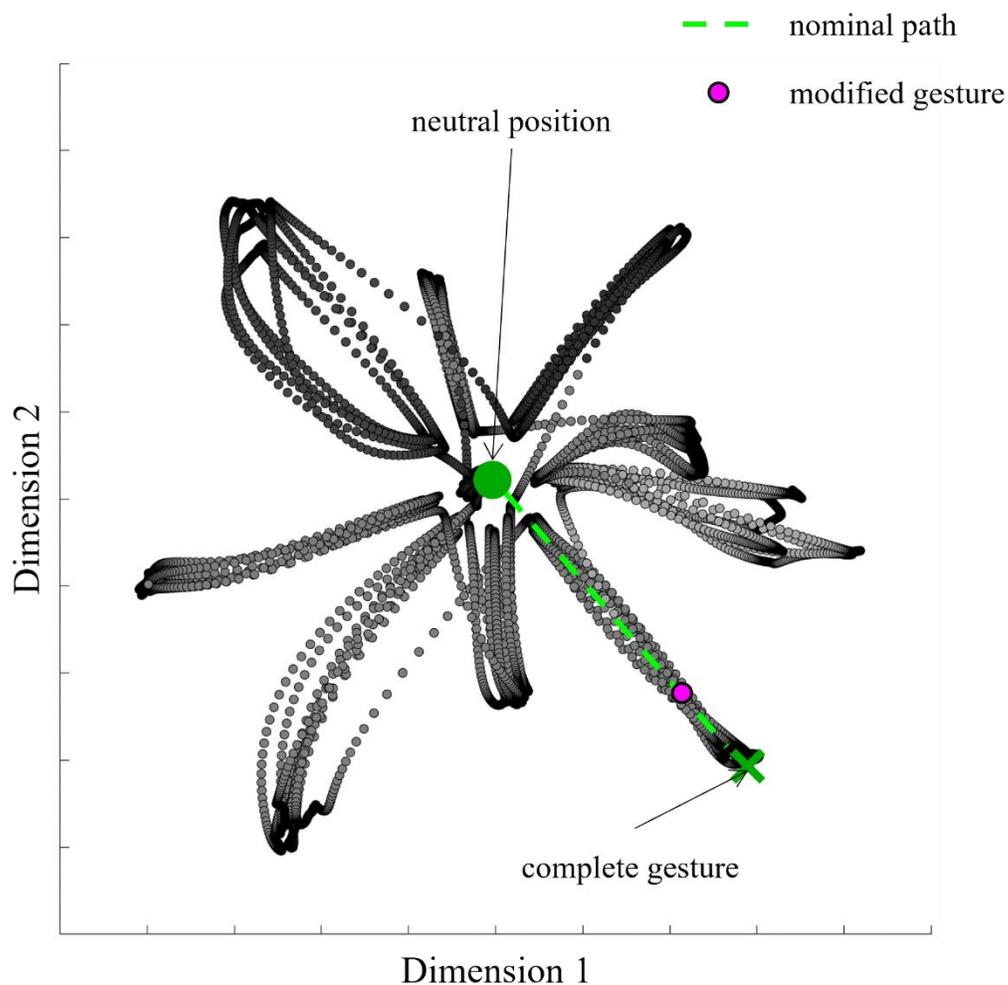


Figure 5-6. Sampling of modified gestures from the latent space. Modified gestures were sampled from 75% of the nominal path between the neutral position and the complete gesture on the latent space.

Study II

To assess how much of the challenge of learning the low-dimensional controller was due to the difficulty of operating the myoelectric interface, we designed Study II, in which the cursor position or the virtual hand gesture were controlled solely by the movement of a computer mouse (*Figure 5-1B*). We hypothesized that when using a computer mouse, one would not have to focus on learning to operate their wrist muscle in a joystick manner, consequently unlearning the natural

ways of using forearm muscles to generate the ASL gestures, and instead would concentrate on learning to operate the virtual hand.

A new group of 22 unimpaired right-handed participants was recruited (12 males, 10 females, 27.4 ± 5.8 years old). All of them were naïve to the controller. As with Study I, the participants were split into two training groups: $2D_{\text{implicit}}$ (11 participants) and $17D$ (11 participants). The rest of the protocol was as in Study I.

During both training and test sessions, the participants did not have any visual feedback of the location of their mouse cursor. This was done in order to closely mimic the conditions of the myoelectric interface in Study I. Upon completion of every trial (during training and test sessions), the participants were asked to return to neutral position (*i.e.*, the center of the 2D control space) by releasing the mouse button, which mimicked muscle relaxation during Study I.

Study III

To explore the importance of explicitly stating the underlying dimensionality of the controller and its connection to the presented high-dimensional task, we recruited a total of seven participants (27.6 ± 6.9 years old), who were naïve to the controller (*Figure 5-1C*). The training group that they were assigned to was $2D_{\text{explicit}}$ and operated the controller via the mouse interface. For analysis purposes, their results were compared to 11 participants from the $2D_{\text{implicit}}$ group in Study II.

During their $2D_{\text{explicit}}$ task in *Train1*, they always observed both hands (the matching and the controlled ones). There was a 2D plane presented between the two virtual hands. The plane was a visual representation of the underlying control space. On the plane, there were eight red targets presented at all times, which corresponded to the 2D position of the eight gestures the participants

were required to learn to recreate during training. A blue cursor corresponded to the 2D location of their controller, which, in turn, mapped into the 17D hand that the participants observed on the right (*Figure 5-5C*).

After hearing a sound cue, the hand on the left showed a new gesture, and the participants had 60s to determine which of the eight red targets produced the desired gesture in the hand on the right. Once the two hands were within an acceptable range, they both turned yellow. Holding the controller within the acceptable range (same as in Studies I and II) for 0.75s led to a successful gesture-matching, turning both hands green.

Neither the blue cursor nor the red targets provided any visual feedback on the correctness of the gesture-matching. Only the target and the controlled hands provided visual feedback by turning yellow (within the acceptable range) or green (correct) through the session. This, in turn, forced the participants to pay attention to the hand gestures as well as cursor/target location on the 2D plane, thus creating a more explicit connection between the 2D planar task and the 17D virtual hand gesture.

The rest of the training and test sessions was as in Study II.

Outcome Measures

Performance between and within the three groups in each study was assessed with the following metrics:

Success Rate

Success rate measured the percentage of successful trials performed in a single session.

Adjusted Reach Time (ART)

Adjusted reach time was defined as the time taken to complete a hand gesture match (in the 17D task) or target reach (in the 2D task). For every missed trial, the ART of the trial was set to the timeout value (60s). ART was only calculated for training trials.

Adjusted Path Efficiency (APE)

Adjusted path efficiency was a measure of straightness of the path taken to either reach the 2D target or match the 17D gesture. It was calculated using *Equation 10*, where d_{travel} was the length of the path covered by the cursor to reach the target/gesture and $d_{nominal}$ was the nominal distance between the central and the final target/gesture.

$$PE = \frac{d_{ideal}}{d_{travel}} * 100\% \quad (10)$$

Similar to ART, for every missed trial, the APE of the trial was set to the lowest possible value of 0%.

Learning Rate

Learning rate was calculated by fitting an exponential function (*Equation 11*) to the curve of ART over all training repetitions, where the participants performed 17D gesture-matching tasks. There, y was the average ART value per repetition, x was the repetition number, and b was the measure of the learning rate.

$$y = a * e^{-b*x} + c \quad (11)$$

Statistical Analysis

For statistical analysis, we used MATLAB Statistics Toolbox functions (*MathWorks, Natick, MA, USA*). Anderson-Darling Test was used to determine the normality of the data (Anderson and

Darling, 1954). Since all data were determined to be non-Gaussian, we used non-parametric tests for statistical analysis.

We evaluated differences *within* and *across* groups were tested on the following metrics:

- average success rates per training or test sessions
- average ART for the first or last repetition of the 2D_{implicit/explicit} or 17D tasks
- average APE for the first or last repetition of the 2D_{implicit/explicit} or 17D tasks

Differences *across* the groups were determined by applying Wilcoxon Rank Sum Test while differences *within* the groups were tested using Wilcoxon Sign Rank Test (Wilcoxon, 1945). In all our analyses, the threshold for significance was set to 0.05.

Results

Study I

Success Rate

The 2D_{implicit} group completed the 2D_{implicit} task with a 100% success rate (*Figure 5-7A*). When switching to the gesture-matching task in *Train1*, its success rate decreased significantly to an average of 86.8%. Its average success rate during *Train2* was 91.8% - although not a significant improvement from the first training session ($p = 0.16$). The success rates during *Test1* and *Test2* for the group were an average of 33.6% and 42.9%, respectively.

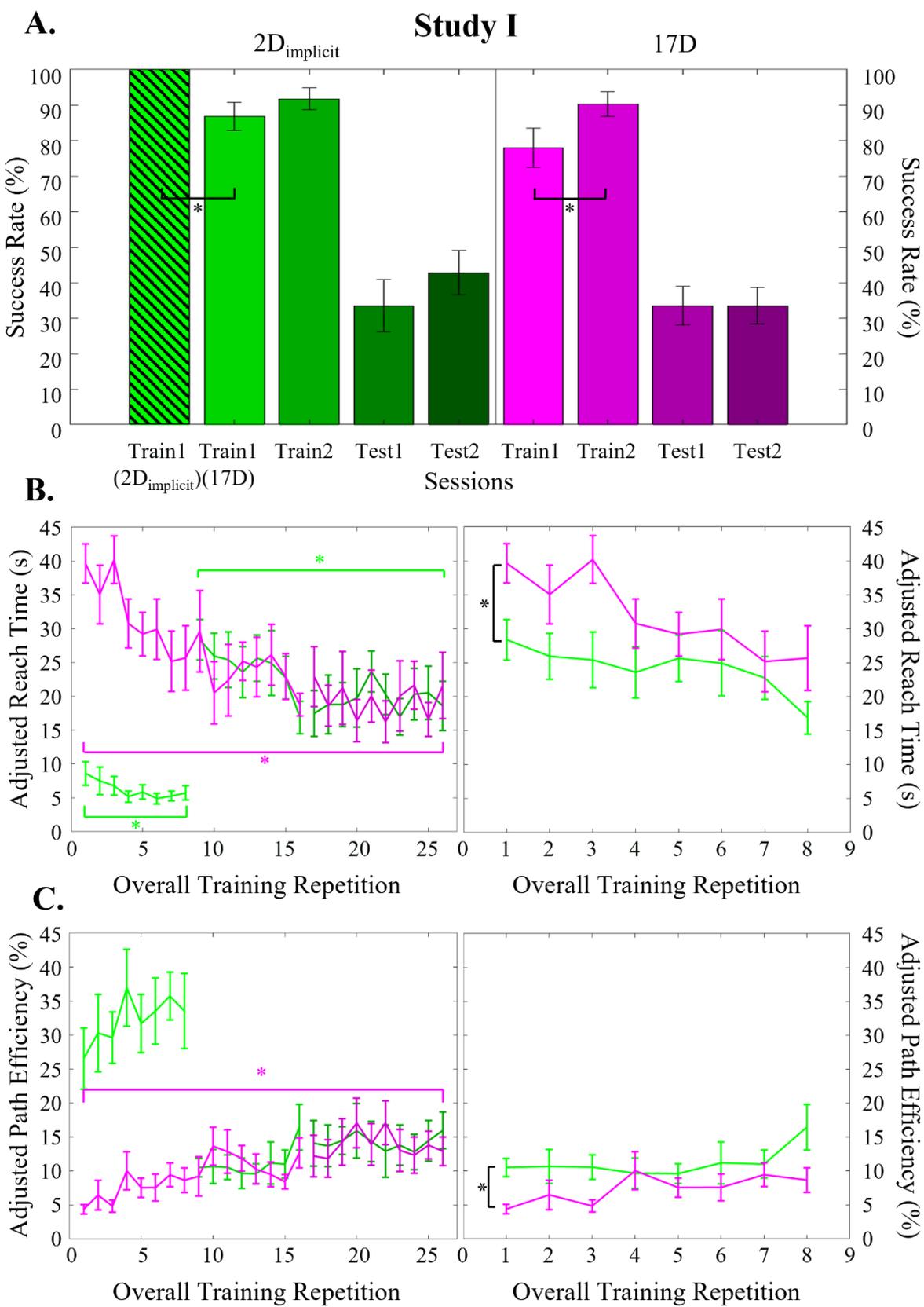


Figure 5-7. Outcome measures of Study I. **(A)** Average success rate for each session across groups. **(B, left)** Average adjusted reach time (ART) for every repetition across all training sessions for each group. **(B, right)** Average ART for the first 8 repetitions of the 17D task for each group. **(C, left)** Average adjusted path efficiency (APE) for every repetition across all training sessions for each group. **(C, right)** Average APE for the first 8 repetitions of the 17D task for each group. Green colors represent different training/test sessions of the 2D_{implicit} group. Magenta colors represent different training/test sessions of the 17D group. Error bars represent standard errors. Black asterisks (*) signify statistical differences *between* the groups. Magenta asterisks (*) represent statistical differences *within* the 17D group. Green asterisks (*) represent statistical differences *within* the 2D_{implicit} group.

The 17D group had a significant improvement between *Train1* and *Train2*, increasing its success rate from an average of 78.0% to 90.4%, respectively. The success rates during *Test1* and *Test2* were both 33.6%.

No statistical difference was observed across the two groups.

Adjusted Reach Time

The 2D_{implicit} group significantly improved its ART during the 2D task – from an average of 8.6s to 5.7s (*Figure 5-7B, left*). It also had a significant improvement in reach time for the 17D task – ART decreased from an average of 28.4s to an average of 18.6s by the end of training.

The 17D group also significantly decreased its ART from an average of 39.7s to an average of 21.6s.

When both groups just completed the first repetition of the 17D task (*Figure 5-7B, right*), we found a significant difference (39.7s for the 17D group and 28.4s for the 2D_{implicit} group). At the end of the first eight repetition of the 17D task, however, there was no significant difference between the two groups anymore.

Fitting an exponential curve to the reaching time values, described in Equation 11, we obtain the learning rate of 0.53 ($R_{adj}^2 = 0.85$) for the 2D task. The learning rate decreases to 0.22 ($R_{adj}^2 = 0.60$) when the 2D_{explicit} group switched to the 17D task. For the 17D group, the learning rate for the 17D task is 0.13 ($R_{adj}^2 = 0.83$).

Adjusted Path Efficiency

The 2D_{implicit} group increases its APE from 26.5% to 33.5% during the 2D task (*Figure 5-7C, left*). The improvement is not statistically significant ($p = 0.11$). When switching to the 17D, the group completes the first repetition of target with an average APE of 10.5% and increases the APE to an average of 16.0% by the end of the training. Once again, the increase is not statistically significant ($p = 0.08$).

On the contrary, the 17D group was able to significantly increase its APE over the course of the 17D task training – from an average of 4.4% to 12.9%.

When comparing what APE values both groups start the 17D tasks for the first time, the values are 10.5% and 4.4% for the 2D_{implicit} and the 17D groups, respectively (*Figure 5-7C, right*). The difference between the two is statistically significant although it diminishes by the time the first eight repetitions of the 17D task are completed ($p = 0.1$).

Study II

Success Rate

The 2D_{implicit} group completed the 2D task with high success rate of 99.3% (*Figure 5-8A*). When switching to the 17D task in *Train1*, the success rate significantly decreased to 86.6%. The success rate for *Train2* was 91.3% - a statistically significant improvement from *Train1*. During test

sessions, the group successfully completed 46.1% of *Test1* and 54.1% of *Test2*. The difference between the test session is statistically significant.

Study II

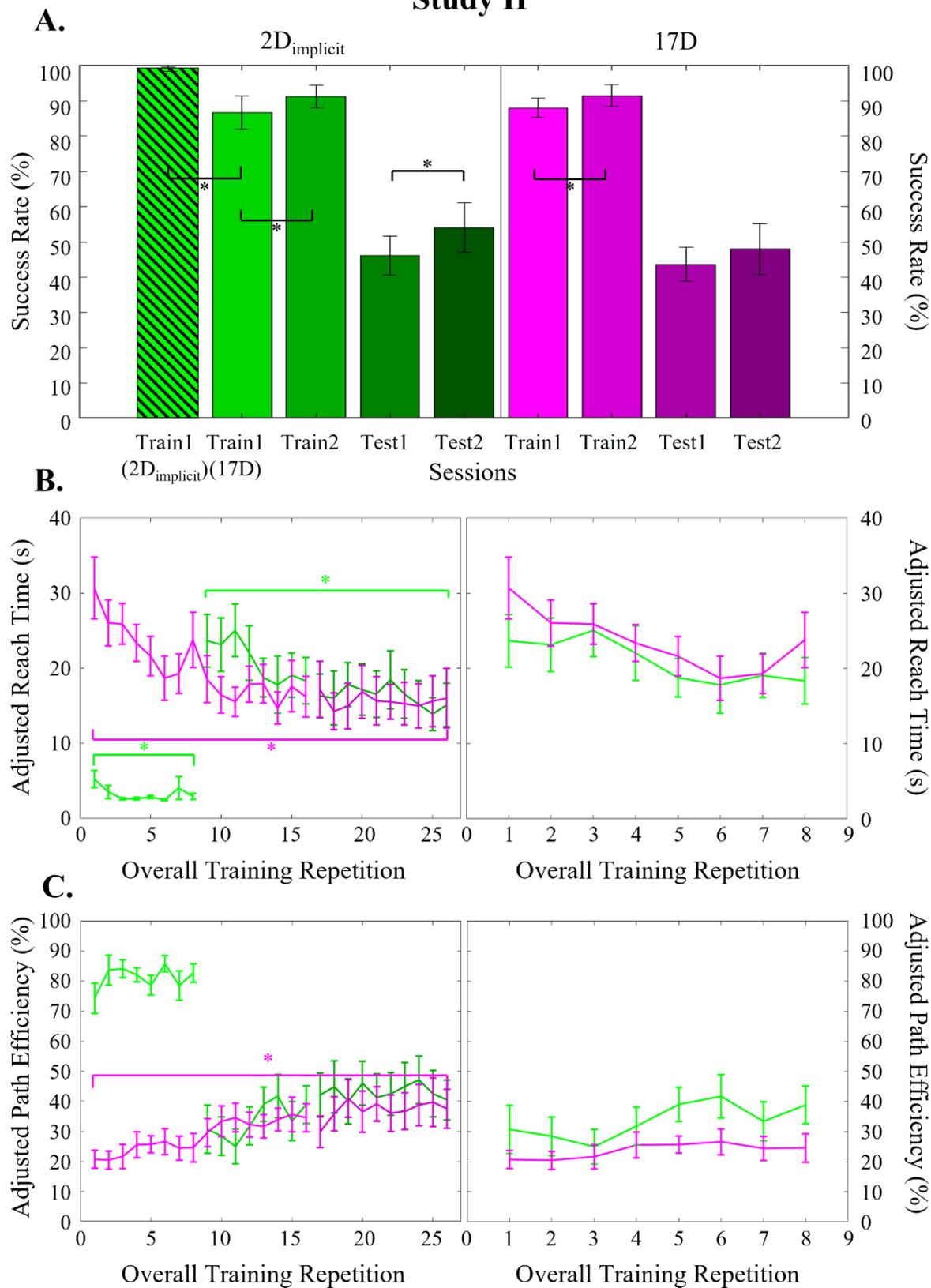


Figure 5-8. Outcome measures of Study II. **(A)** Average success rate for each session across groups. **(B, left)** Average adjusted reach time (ART) for every repetition across all training sessions for each group. **(B, right)** Average ART for the first 8 repetitions of the 17D task for each group. **(C, left)** Average adjusted path efficiency (APE) for every repetition across all training sessions for each group. **(C, right)** Average adjusted APE for the first 8 repetitions of the 17D task for each group. Green colors represent different training/test sessions of the 2D_{implicit} group. Magenta colors represent different training/test sessions of the 17D group. Error bars represent standard errors. Magenta asterisks (*) represent statistical differences *within* the 17D group. Green asterisks (*) represent statistical differences *within* the 2D_{implicit} group.

The 17D group successfully completed 88% of *Train1* and 91.5% of *Train2*. The difference between the training sessions is statistically significant. Both *Test1* and *Test2* were successfully completed at a rate of 43.6% and 48.0%, respectively, although the improvement is not statistically significant.

Adjusted Reach Time

The participants in the 2D_{implicit} group were able to significantly improve their average ART from the first to the last repetition in the 2D_{implicit} task (from 5.3s to 2.9s, respectively) (*Figure 5-8B, left*). The group was able to significantly improve its ART for the 17D task as well – from 23.6s for the first repetition and 15.1s for the last repetition during training.

Similarly, the 17D group had a significant improvement in its average ART value – from 30.7s to 16.0s.

The difference between the two groups at the end of the training sessions was not statistically significant ($p = 0.79$). When considering the initial repetition of the 17D task for both groups, there was no significant difference between their values ($p = 0.29$) (*Figure 5-8B, right*).

The 2D_{implicit} group had a learning rate of 1.7 ($R_{adj}^2 = 0.62$) for the 2D task. For the 17D task, the learning rate was 0.15 ($R_{adj}^2 = 0.80$). The 17D group exhibited a learning rate of 0.20 ($R_{adj}^2 = 0.88$) during training.

Adjusted Path Efficiency

The increase of the APE values for the 2D_{implicit} group during the 2D_{implicit} task was not statistically significant ($p = 0.37$): from 74.3% to 82.7% (*Figure 5-8C, left*). Similarly, the increase in the APE from 30.7% to 40.0% during the 17D task was not statistically significant ($p = 0.10$).

As in Study I, only the 17D group significantly increased its average APE value over the course of the training repetitions – from 20.7% to 37.5%.

Comparing the first eight repetitions of the gesture-matching task across both groups did not reveal any statistical significance (*Figure 5-8C, right*).

Study III

Success Rate

The 2D_{explicit} group exhibited high average success rates during training sessions (100%, 99.1%, and 99.8% for *Train1* (2D_{explicit} task), *Train1* (17D task), and *Train2*, respectively) (*Figure 5-9A*). The average success rates in the 17D training sessions of the 2D_{explicit} group were significantly higher than those of the 2D_{implicit} group. Average success rate during *Test2* was significantly higher for 2D_{explicit} group (77.5%) than the 2D_{implicit} one (54.1%). The difference between the groups during *Test1* was not statistically significant.

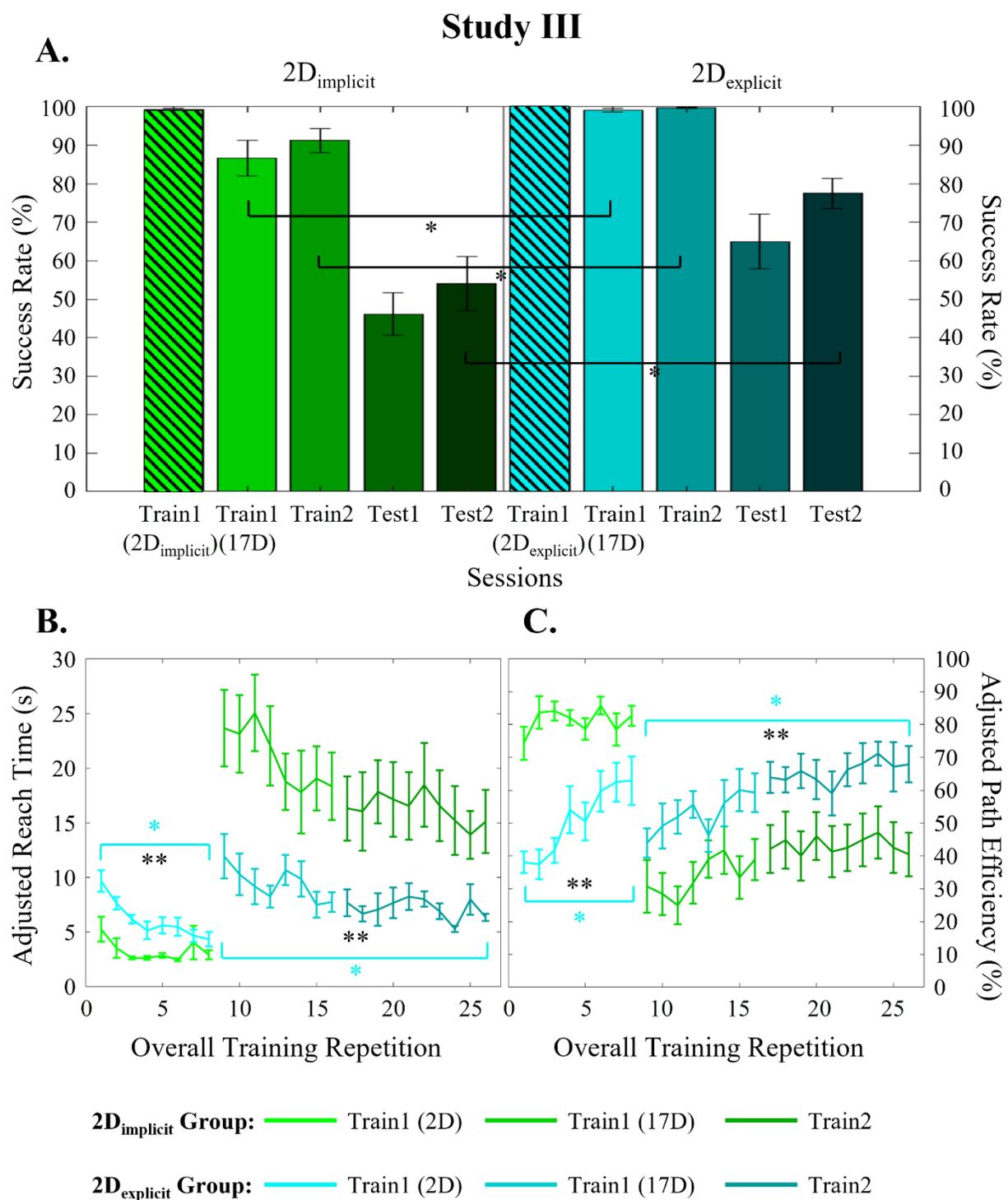


Figure 5-9. Outcome measures of Study III compared to Study II ($2D_{\text{implicit}}$ group only). (A) Average success rate for each session across groups. (B) Average adjusted reach time (ART) for every repetition across all training sessions for each group. (C) Average adjusted path efficiency

(APE) for every repetition across all training sessions for each group. Green lines represent different training/test sessions of the $2D_{\text{implicit}}$ group. Blue lines represent different training/test sessions of the $2D_{\text{explicit}}$ group. Blue asterisks (*) represent statistical significance within the $2D_{\text{explicit}}$ group. Black double asterisks (**) represent statistical significance between the $2D_{\text{explicit}}$ and $2D_{\text{implicit}}$ groups.

Adjusted Reach Time

The participants in the $2D_{\text{explicit}}$ group significantly decreased their average ART during the $2D_{\text{explicit}}$ task – from 9.7s to 4.4s (*Figure 5-9B*). Once the visual presentation of the targets, cursor, and the 2D plane was switched off, the participants completed the first repetition of the 17D task within 12.0s on average and were able to significantly reduce the ART to at an average of 6.4s by the end of the training.

During the $2D_{\text{implicit/explicit}}$ tasks alone, the $2D_{\text{implicit}}$ group reached targets significantly faster than the $2D_{\text{explicit}}$ group at the beginning and the end of the 2D task trials.

However, the $2D_{\text{explicit}}$ group began the 17D task at a significantly lower ART value (12.0s) than the $2D_{\text{implicit}}$ group (23.7s). At the end of the training, the $2D_{\text{explicit}}$ group was also successfully matching gestures at a significantly faster ART (6.4s) than the $2D_{\text{implicit}}$ group (15.1s).

The $2D_{\text{explicit}}$ group exhibited a learning rate of 0.54 ($R_{adj}^2 = 0.95$) during the $2D_{\text{explicit}}$ task, which is significantly lower than that of the $2D_{\text{implicit}}$ group during the $2D_{\text{implicit}}$ task. Its learning rate for the entirety of the 17D task was also lower at the value of 0.18 ($R_{adj}^2 = 0.63$) than that of the $2D_{\text{implicit}}$ group.

Adjusted Path Efficiency

The $2D_{\text{explicit}}$ group significantly improved its APE during both the $2D_{\text{explicit}}$ and the 17D tasks (*Figure 5-9C*). During the $2D_{\text{explicit}}$ task, the improvement is from an average of 38.1% to 62.9%.

During the 17D task, the improvement is from an average of 43.9% to 67.9%. In addition, the final APE of the $2D_{\text{explicit}}$ group at the end of the 17D task (67.9%) was similar to the level that the group was able to achieve by the end of the $2D_{\text{explicit}}$ task (62.9%).

When considering the $2D_{\text{explicit/implicit}}$ tasks, the $2D_{\text{explicit}}$ group had consistently lower APE values than the $2D_{\text{implicit}}$ group. When switching to the 17D task, the difference of average APE values between the $2D_{\text{explicit}}$ and $2D_{\text{implicit}}$ groups was not statistically significant ($p = 0.13$). However, by the end of the training, the $2D_{\text{explicit}}$ group was able to perform significantly more efficient reaches (67.9%) than the $2D_{\text{implicit}}$ group (40.0%).

Discussion

Through the series of the developed studies, we explored various aspects that might affect the user's ability to learn the low-dimensional controller. Here, we highlighted the challenges of operating the controller via a myoelectric interface and determined the importance of providing the user with an explicit connection between the underlying low-dimensional control space and the presented high-dimensional task. In addition, we discuss the potential differences in learning linear and nonlinear controllers and compare the user performances obtained during our studies with other low-dimensional (PCA-based) controllers and assess the potential reasons for the differences in the results. Lastly, we discuss limitations of the developed studies and assess the potential implications of our findings for prosthetic users.

Difficulty of Myoelectric Control

The notion that the users can learn any mappings that you provide them (intuitive or not) has been supported by multiple studies (Liu and Scheidt, 2008; Radhakrishnan et al., 2008; Ison and Artemiadis, 2015; Zhou et al., 2019; Dyson et al., 2020). Oftentimes, in cases when a developed

controller is operated via a novel (*e.g.*, myoelectric) interface, we forget to differentiate between the processes of learning the interface and learning the controller. Nonetheless, when both the controller (*i.e.*, the map) and the interface are novel, as it was in our case, the user is required to learn both, oftentimes, simultaneously.

One of the factors that we aimed to explore in this paper was understanding how much of the difficulty of learning the low-dimensional controller was due to the challenges of operating a myoelectric interface itself. The main difficulty that the participants in Study I may have experienced with the myoelectric interface could be the mismatch between the natural way of using forearm muscles to create gestures and the new way they were required to learn to use their wrist to operate the virtual hand. The new way was evidently unnatural and required an additional learning on top of controller learning

The main significant difference between Studies I and II was that the participants in both groups (17D and 2D_{implicit}) were able to reach targets or match gestures with a mouse interface in a significantly more efficient way (higher APE) using the mouse interface than the myoelectric interface. This observation appears to be self-evident – people perform straighter reaches by moving the mouse with their hand rather than trying to independently activate their muscles in an unnatural way. However, no significant differences were observed between the final ART values across both groups in Studies I and II, suggesting that the myoelectric interface was learnable and was not the main reason for poor performances across the Study I participants. When observing the learning rates across Studies I and II, we could see that they were comparable, suggesting that even when the operating interface was familiar to the user, a great factor of learning inhibition was due to the difficulty of the novel controller itself.

Effects of Learning Paradigms

Having identified the difficulty that the participants had in learning the low-dimensional controller, we explored how different learning paradigms inhibited or aided learning of the controller.

Let us first consider the 2D training provided to the $2D_{\text{implicit}}$ group. There, the relationship between the underlying dimensionality of the controller and the gesture-matching task that followed the target reaching was not explicitly made. Presentations of a single target at a time along with smaller avatars of the controlled and matching hands, most likely, encouraged the participants to focus on the target location rather than the generated gesture. This observation is supported by the fact that the $2D_{\text{implicit}}$ group did not outperform the 17D group during gesture-matching at the end of *Train2*.

An interesting observation can be made when looking at the results of Study I: when operating a myoelectric interface, participants that completed eight repetitions of $2D_{\text{implicit}}$ task, began the gesture-matching task at a significantly faster ART than the 17D group completing their first repetition of the 17D task. One may think that the initial training in 2D helped the participants perform their gesture matches faster. However, it is important to understand that the $2D_{\text{implicit}}$ task did not only provide training on the controller but on the myoelectric interface as well.

When adjusting for the eight repetitions of either the 17D or $2D_{\text{implicit}}$ task, both groups completed gesture matching at comparable ARTs. This points to the fact that the initial training in 2D worked as well as (but not better) as having the initial training on the 17D task. The same effects are observed when looking at APE during Study I – the $2D_{\text{implicit}}$ group had a significant superiority over the 17D group during its first repetition of the 17D task. But when adjusted for the training time on the myoelectric interface, the superiority of the $2D_{\text{implicit}}$ group disappeared and the participants across both groups performed equally well.

When looking at the results of Study II, we do not observe a significant difference between the first repetitions of the 17D task in neither of the groups (in terms of both ART and APE). It is important to consider that in Study II, the participants controlled the interface using a computer mouse – something that they were assumed to have full proficiency of as the mouse moved as it would in daily activities when operating a computer. This strengthens the point that the $2D_{\text{implicit}}$ training mostly affected the improvement in the myoelectric control proficiency for the participants in Study I who were naïve to the interface.

When considering the learning rates during the $2D_{\text{implicit}}$ task between Studies I and II, we can see that, once operating a more familiar mouse interface (Study II), the participants exhibit a significantly faster learning rate ($LR = 2.1$) than when doing so with an unfamiliar myoelectric interface (Study I, $LR = 0.53$). This strongly suggests that the slower learning rate was attributed to the difficulty due to the unfamiliarity of the myoelectric interface.

When designing Study III, we hypothesized that learning that took place during the 2D task did not provide the $2D_{\text{implicit}}$ group with an explicit understanding of the underlying dimensionality of the control space. Instead, it only trained them to perform abstract (as they appeared to the participants) movements on a 2D plane. As a result of this observation, what we wanted to emphasize in the following study was as follows: locations on a 2D plane had a direct relationship to the eight gestures presented in the matching hand and to perform proper gesture matching, one must reach the required target in 2D, which would then be reconstructed to full hand kinematics. We hypothesized that once this relationship was understood, it would become a pure memorization problem of the gesture locations on the 2D plane.

The presentation of all eight targets without error feedback directly on the targets themselves forced the participants to pay attention not only to the target locations but at the desired hand gestures. In addition, in Study III, the participants clearly observed how their movement on the 2D plane related to the generated gestures as the hands were now more visually accentuated and significantly larger than those presented in Studies I and II.

The results of Study III suggest that establishing an explicit connection between the underlying dimensionality of the controller and the presented hand gestures is essential when training a low-dimensional controller. Without a clear understanding that what they truly did was reach targets on a 2D plane, the participants were unable to achieve high enough proficiency to perform fast gesture matches.

Linear vs Nonlinear Postural Controllers

We hypothesize that the explicit understanding of the controller-task relationship would be essential when teaching users novel controllers developed using *nonlinear* DR methods; however, such understanding might not be relevant in cases where PCA (a *linear* method) has been used. The reason for that would be that the latent space created by PCA follows the superposition principle. This means that new gestures could be superimposed from other gestures on the 2D plane. A good example of that would be if one dimension of the latent space solely controls the opening and closing of the thumb, while the other one flexes and extends the other four fingers, gestures generated within the space would be linear combinations of these two dimensions.

The superposition principle does not hold in latent spaces generated by nonlinear systems. And while gestures that are close to each other kinematically, appear closer to each other on the latent space encoded by the AE (Portnova-Fahreeva et al., 2020), nonlinear maps might be harder for

users to interpolate from. Nonetheless, the differences in learning linear and nonlinear maps have not been explored in this paper, but we suggest that this might be an interesting route to investigate in future experiments.

One of the major outcomes of the studies described in this paper was the application of nonlinear DR methods, such as AEs, for the development of a nonlinear postural controller, in which complex kinematics of a virtual hand with 17DOFs were extracted from the position on the 2D plane. In the past, linear controllers have been developed, in which the dimensionality of hand kinematics during grasping was reduced using PCA (Magenes et al., 2008; Matrone et al., 2010; Matrone et al., 2012; Belter et al., 2013; Segil, 2013; Segil and Controzzi, 2014; Segil, 2015).

In the studies where linear postural controller was validated with a myoelectric interface on actual participants (Matrone et al., 2012; Segil and Controzzi, 2014; Segil, 2015), the average movement times (time to successfully reach but not hold the hand in a correct grasp) were between 3s and 5s. The results are comparable to those produced in our Study III; however, differ in the interface used to perform the movement. When compared to the results of our myoelectric interface study with longer exploration windows (Study I), movement times using the PCA-based controller were significantly lower than those using the AE-based controller.

Explanations for the discrepancies in the results could be due to the major differences in the controller schemes and protocols. First of all, the dimensionality of the output system with a PCA-based controller was 5 – 6 DOFs, in contrast to the 17 DOFs controlled in this paper, resulting in a more intricate but, most likely, complex control. In addition, in one of the aforementioned studies (Matrone et al., 2012), the participants were only required to learn to create three grasps in

comparison to learning eight unique ASL gestures in our studies. Another study (Segil, 2015) implemented potential fields that “snapped” the virtual hand in correct postures when the control cursor was close enough to the target posture on the 2D plane. That allowed for a simpler control and alleviated the challenges experienced during myoelectric control due to the noisy nature of EMG.

By evaluating the results of the studies in which users operated PCA-based controllers, we can observe that the users were able to achieve high performance in comparison to those achieved in our studies using a myoelectric interface. However, following the results of Study III, in which we discover the most appropriate form of training paradigm for the AE-based controller, we hypothesize that high performance could also be achieved with the nonlinear postural controller, assuming that the users are able to learn to operate an EMG interface as well as they operated a mouse interface. As a result, we suggest that nonlinear postural controllers could still be a viable option for complex prosthetic control allowing for more precise dimensionality-reduction of intricate hand kinematics than what could be achieved by PCA.

Limitations

One of the major limitations of our studies was the design of the test sessions with very short window to perform reaches. Considering that the average ART during training sessions in Studies I and II was significantly higher than the time allowed for a successful reach during test, the participants were set up for failure, which explains the low success rates during test. In Study III, the average ART at the end of the training sessions is similar to the time allowed for a successful gesture-matching in test, which explains a significantly higher success rates during test sessions for the 2D_{explicit} group.

Another limitation was a relatively small sample size for some of the studies, reducing the statistical power of the experiments. Considering that data collection took place during the times when the entire world was going to the pandemic, participant recruitment proved to be difficult. Studies that could be conducted remotely yielded a higher number of participants.

Applicability to the Prosthetic Users

When designing these studies, the end-user group that we considered were upper-limb amputees that utilize prosthetic hands in their daily living. Although the studies were performed on unimpaired individuals, they highlighted the possibility of using nonlinear controllers for the purpose of manipulating a myoelectric hand prosthesis. The myoelectric interface that we designed for Study I employed four wrist signals to operate on the 2D latent space. And although an upper-limb amputee might not have those wrist muscles, other more distal locations can be chosen to obtain clean signals to control a location of a 2D cursor, which, in turn, operates the hand. The main advantage of our controller is that it does not require a large number of signals (only enough to operate the cursor on the 2D plane) to control a hand with a large number of DOFs. One does not even need to limit themselves to the EMG system – a 2D control signals can be obtained from a simple interface like Internal Measurement Units (IMUs). For example, an IMU can be placed on the user's shoulders, consequently, controlling the posture of the prosthetic hand. In the past, IMUs have been widely used to operate a low-dimensional controller (Thorp et al., 2015; Seáñez-González et al., 2016; Abdollahi et al., 2017; Pierella et al., 2017; Rizzoglio et al., 2020). Thus, nonlinear AE-based controllers, such as the one we developed for our studies, can be a versatile and modular solution for controlling complex upper-limb prosthetic devices via low-dimensional interfaces.

Chapter 6 Three Challenges of Myoelectric Low-Dimensional Control

Introduction

Dimensionality-reduction methods are important tools for uncovering the underlying structures of complex signals. In a biological context, they have been used to understand the control strategy of arm and leg muscles (Ting, 2007; Ting et al., 2012; De Groote et al., 2014; Phinyomark et al., 2018; Junior et al., 2020) as well as to uncover the underlying representations of complex hand kinematics during object grasping and sign-language gesturing (Santello et al., 1998; Todorov and Ghahramani, 2004; Portnova-Fahreeva et al., 2020). With the ability to compress high-dimensional signals into a lower-dimensional, latent, space, dimensionality-reduction methods serve as a powerful platform for control of robotic devices (*e.g.*, prosthetic hands) with a large number of degrees of freedom (DOFs).

Previously, our group has focused on applying Autoencoders (AEs), one of many dimensionality-reduction methods, to hand kinematics as a way of simplifying the control of a virtual hand with 17 DOFs (see *Chapter 5*). AEs are a computational architecture that aims to minimize the difference between the input and the output reconstructed from a latent space of lower dimension. In *Chapter 5*, we encoded kinematics of a biological hand into a lower-dimensional set of commands, allowing the user to operate a virtual hand via a 2D interface. However, dimensionality reduction was only used at the output end to obtain reconstructed hand kinematics. Like in many other robotic interfaces, such as hand prostheses, input signals through which the user operates the interface are just as important as the output of the system (*i.e.*, joint angles). In case of myoelectric prostheses, the input is comprised of electromyographic (EMG) signals.

As we have discovered in a previous study (see *Chapter 5*), a challenge of operating myoelectric interfaces is related to the difficulty of learning to use muscles in a new and unnatural manner to control an external device. A potential way of alleviating this difficulty may lie in applying a dimensionality-reduction method to muscle signals. Not only might it uncover the hidden *natural* muscle combinations for control, but also eliminate the dimensionality mismatch between the input (control) and the output (controlled) signals.

For EMG signals, a dimensionality-reduction method of use has mainly been Nonlinear Matrix Factorization (NNMF) (Tresch et al., 2006; Lambert-Shirzad and Van der Loos, 2017; Rabbi et al., 2020). In these studies, the purpose of dimensionality reduction was to uncover the hidden structure of what was assumed to be muscle control strategy via synergies. However, in such cases, NNMF was not used for control purposes, but for synergy extraction alone.

AEs have a history of use for simplifying control. Vujakija and colleagues (Vujaklija et al., 2018) trained two separate AE models on EMG signals recorded either during wrist flexion/extension or wrist abduction/adduction and used the AE output to control a 2D cursor. The AE that was trained to recognize wrist flexion/extension moved the cursor horizontally while the AE trained on wrist abduction/adduction moved it vertically on a 2D monitor. However, when attempting to apply this method, we found it difficult to locate *independent* muscle signals necessary to differentiate between flexion/extension and abduction/adduction with a conventional surface EMG acquisition system. When we recorded signals during abduction/adduction, muscles involved in flexion/extension were contracting as well, consequently being encoded into the lower-dimensional space of the AE model trained on abduction/adduction. Therefore, an individuated control of two axes on the plane was impossible. Changing the position of electrodes did not

alleviate this issue. Hence, the approach of using separate AEs might not be ideal for prosthetic hand control, considering that EMG signals are acquired via electrodes located inside of a socket, which oftentimes cannot provide for the most accurate location of the muscles due to daily limb volume changes and electrode shift (Sanders et al., 2007; Sanders and Fatone, 2011).

As a result, we set on to explore how applying various dimensionality reduction methods on EMG signals might affect the control space. Here, we looked at Principal Component Analysis (PCA), AE, and variational AE (VAE) and evaluated their applicability for the purpose of a low-dimensional myoelectric controller. We outline different factors, outside of the standard measure of reconstruction error, to be considered when generating a latent space for a control purpose. The three factors identified in this study are latent dimension variance, class separability, and location of muscle resting signals. We evaluated these factors by applying PCA, AE, and VAE to different EMG datasets of various complexity. In addition, we explored the effects of the structure of the input data, such as user's impairment level and signal density (number of acquired EMG signals), on latent-space organization.

Importantly, this study explores the applicability of dimensionality reduction methods to myoelectric signals solely in the context of what has been termed as a postural control (Segil, 2013; 2015; Segil and Huddle, 2016). Postural control is defined as a way of controlling hand kinematics (virtual or prosthetic) via a latent space. There, dimensionality reduction is applied to both the controlling EMG signals and to the controlled hand kinematics, consequently reconstructing hand gestures from a latent space.

Methods

Dimensionality-Reduction Methods

Principal Component Analysis

PCA is a linear method, widely used to reduce the complexity of biological signals. This is done by projecting the input data along latent directions, effectively compressing representations of the input data (*Figure 6-1A*). Data compression by PCA has a few key features: 1) orthogonality of latent dimensions and 2) ability to only account for linear relationships in the input data. Considering that biological signals are oftentimes nonlinear in their nature, this method can be inadequate when applied to hand kinematics or muscle signals.

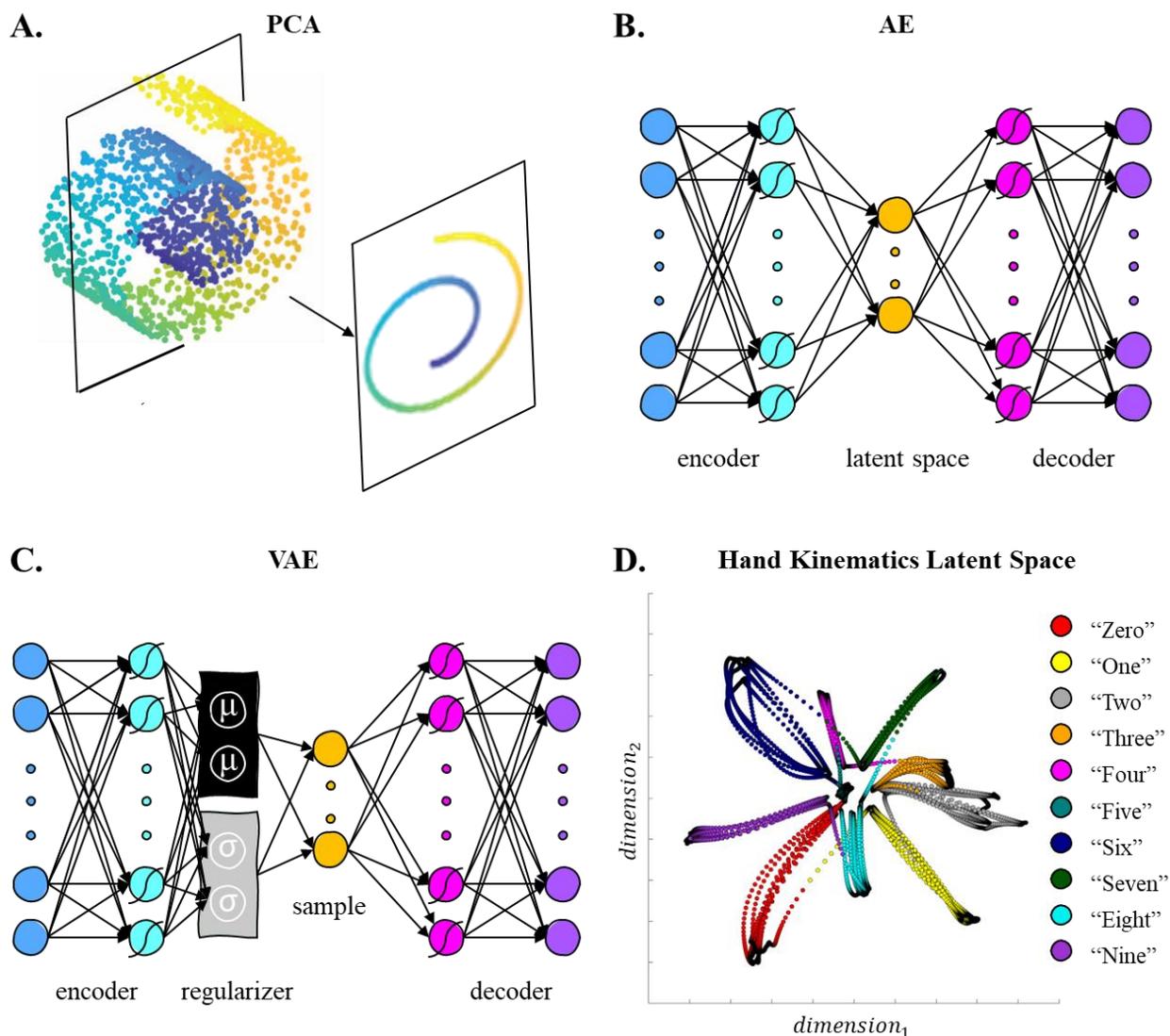


Figure 6-1. (A) Principal component analysis (PCA) achieves dimensionality reduction by finding the directions along which the input data vary the most and projecting the data along them. (B) Autoencoder (AE) network performs dimensionality reduction by compressing the input into the latent space via the encoder and reconstructs the output via the decoder. (C) Variational autoencoder (VAE) network performs dimensionality reduction, similar to AE, and regularizes its encodings distribution during training in order to ensure that its latent space has a target distribution. (D) Latent space of hand kinematics generated by applying a VAE on the data from

a participant performing American Sign Language number gestures. Different colors represent various number gestures.

Autoencoders

AEs are artificial neural networks consisting in the cascade of two components: an encoder that compresses the input to a latent representation, followed by a decoder that expands the latent representation into the output, with the same dimensions as the input (*Figure 6-1B*). The AEs learn to efficiently encode the input data variability within its latent space by minimizing the reconstruction error between the output and the input (*Equation 1*).

$$Loss_{AE} = \|x - \hat{x}\|^2 \quad (1)$$

For an AE to be considered nonlinear, it must contain at least one nonlinear activation function between layers. Although AEs can vary greatly in the complexity of their structure, for the purpose of this study, we use the simple AE structure developed in our previous study (Portnova-Fahreeva et al., 2020). This network includes a total of three hidden layers. The middle layer is a *bottleneck* layer. The first and third hidden layers have a non-linear activation function. The rest of the layers are connected via a linear activation function. The combination of linear and non-linear activation functions had been shown to increase the efficiency with which the network learns linear relationships in the data (Haesloop and Holt, 1990). The structure of each layer is shown in the *Equations 2-5*.

$$layer_1 = \text{nonlin}(X * w_1 + b_1) \quad (2)$$

$$layer_2 = layer_1 * w_2 + b_2 \quad (3)$$

$$layer_3 = \text{nonlin}(layer_2 * w_3 + b_3) \quad (4)$$

$$Y = layer_3 * w_4 + b_4 \quad (5)$$

where X is the normalized input, Y is the normalized output, w_i are the weights, and b_i are the biases of the AE network.

Variational Autoencoders

Despite their strong capabilities of reconstructing biological data with higher precision than their linear counterpart, PCA, standard AEs have a fundamental problem of creating a noncontinuous latent space that may inhibit easy interpolation. Points that are not encoded onto the latent space often reconstruct to unrealistic data. As a result, latent spaces derived from regular AEs may not be the most optimal option for myoelectric prosthetic control.

Here, we explore the use of VAEs on myoelectric data (*Figure 6-1C*). Differently from a standard AE, a regularizing term (Kullback-Liebr Divergence, KLD) (Kullback and Leibler, 1951) is added to the reconstruction error in the VAE cost function, which aims to match the probability distributions of the latent space to that of a given distribution (*Equation 6*). This term is simply a measure of how different the created distribution is from the target distribution.

$$Loss_{VAE} = \|x - \hat{x}\|^2 + \beta * KLD[N(\mu_x, \sigma_x), N(0, I)] \quad (6)$$

By optimizing the two terms of the cost function, the resulting VAE latent space can locally maintain the similarity of nearby encodings via clustering yet is globally densely packed near the latent space origin. In this paper, the distribution of choice is the normal Gaussian distribution ($\mu = 0, \sigma = 1$).

Datasets

For this study, we used three open-source databases that contain myoelectric data recorded during either a variety of common grasps or simple wrist or finger movements. In the databases, EMG was collected with surface electrodes placed on the participants' forearm muscles.

The first database, MeganePro 1(MDS1), contains 12 channels of surface EMG acquired with Delsys (*Delsys Inc, MA, USA*) electrodes and is recorded on 15 transradial amputees and 30 able-bodied individuals (Cognolato et al., 2020). All participants in the database performed grasps in a *static* or *dynamic* way and in either a sitting or a standing position. A static grasp indicates that participants grasped the objects without moving or lifting them. During a dynamic grasp, participants performed a functional task with an object. Both dynamic and static grasps were performed either in a standing or sitting conditions. From this database, we generated two datasets used in this study: 1) *unimpaired dataset* and 2) *amputee dataset*. For the *unimpaired dataset*, a single unimpaired participants (S10) performed object grasping in the *dynamic sit* condition. In the *amputee dataset*, the participant with lower-arm amputation (S101) performed object grasps in the *dynamic sit* condition.

The second database, NinaPro2 (DB2), also contains 12 EMG signals acquired with Delsys electrodes and was recorded on 40 unimpaired individuals (Atzori et al., 2014). From this database, we extracted a *low-density dataset*, in which a single participant (S1) performed flexion, extension, abduction, and adduction movements of the wrist.

The last database, *Hyser*, includes high-density surface EMG acquired with a total of 256 channels from 20 unimpaired participants, recorded on two separate days (Jiang et al., 2021). All participants performed *dynamic* tasks (1s duration, from relaxing state to the required gesture) and

maintenance tasks (4s duration, from a relaxed state to the required gesture followed with maintenance at that gesture). From this database, we created a *high-density dataset*, in which a single participant (S1) performed wrist flexion, extension, abduction, and adduction with fingers fully extended. This dataset was used for comparative purposes with the *low-density dataset*, so the movements were performed in the *maintenance* manner. The dataset includes only the movements performed in the first session.

Pre-Processing

EMG data in MDS1, DB2, and Hyser databases were collected at sampling rates of 1926Hz, 2000 Hz, and 2048Hz, respectively. Prior to performing dimensionality reduction on the data, the created datasets were run through a Bandpass filter (450Hz and 30Hz). The signals were then rectified, and a Low-Pass filter at 2Hz is applied to create a less noisy input for the dimensionality-reduction algorithms. Data from MDS1, DB2, and Hyser databases were downsampled to 107Hz, 100Hz, and 102.4Hz, respectively, to decrease the dimensionality-reduction model training times.

If the dataset contains multiple repetitions of the same grasp type/movement, all repetitions are included in the input data. Both datasets are labeled for the type of grasp/movement the participant performed.

The datasets are randomly split into training (80%) and testing (20%) using a holdout method (Oxford and Daniel, 2001). Training samples are used to generate a dimensionality-reduction model and the outcome measures are calculated on the test samples. Latent spaces are also constructed from the test samples.

To map the location of the resting signal encoded onto the latent space, resting EMG signals, EMG_{rest} , are extracted for *unimpaired*, *amputee*, and *low-density datasets*. The *high-density dataset* does not include any signal data with the forearm muscles completely relaxed, so EMG_{rest} is assumed to be an array of values with $\mu = 0, \sigma = 0.001$ with a dimension equal to the number of collected channels ($n = 256$).

Since the high-density dataset does not contain any recordings of resting muscles, we recreated EMG_{rest} by using a 100×256 matrix of randomly generated numbers on a normal distribution ($\mu = 0, \sigma = 0.01$) and rectified it (Equation 7).

$$EMG_{rest,i} = \left| \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \right| \quad (7)$$

Hyperparameter Tuning

To ensure model convergence, we performed hyperparameter tuning on AE and VAE networks via a five-fold cross-validation (Stone, 1974). For the AE, we evaluated the effects of the learning rate (lr) on the reconstruction error via root-mean-squared error (RMSE). For the VAE, we explored the effects of lr and the weighting factor (β) of the regularization term in the loss function. We did so by evaluating the effects of the explored factor on both RMSE and KLD. *Sigmoid* was the function of choice for nonlinear layers in the network as it matches the range of filtered myoelectric signals, which are always positive. The seed parameter was randomly chosen for each run.

Each dataset was individually tuned for AE and VAE networks. The best-performing hyperparameters are shown in *Table 1*, where lr is a learning rate, n is the number of steps, and β is the weight of the KLD term in the VAE cost function.

Table 6-1. Hyperparameters of the autoencoder (AE) and variational autoencoder (VAE) networks found from hyperparameter tuning on the unimpaired, amputee, high-density, and low-density datasets. lr is the learning rate, n is the number of steps, β is the weighting term on the Kullback-Liebr Divergence term in the VAE cost function.

	AE	VAE
Unimpaired	$lr = 0.01, n = 5,000$	$lr = 0.01, n = 5,000, \beta = 0.0005$
Amputee	$lr = 0.01, n = 5,000$	$lr = 0.01, n = 5,000, \beta = 0.0005$
High-Density	$lr = 0.0005, n = 5,000$	$lr = 0.0005, n = 5,000, \beta = 0.00075$
Low-Density	$lr = 0.005, n = 5,000$	$lr = 0.005, n = 5,000, \beta = 0.0005$

Latent Space of Hand Kinematics

The importance of creating a usable latent space from EMG signals is evident when considering the mapping between the latent space of EMG to that of hand kinematics. For this, we assume that the most informative EMG latent space can be projected to the hand kinematics latent space via a 1:1 mapping, requiring nothing more than a direct transformation between the two. This is where the importance of the developed latent space factors come into play. The hand kinematics latent space is derived by training a VAE on hand kinematics data an individual performing American Sign Language gestures (*Figure 6-1D*).

Latent Space Factors

In the most conventional application of dimensionality-reduction methods, reconstruction power, usually represented by a Variance Accounted For, has been the main metric for determining the appropriateness of the method in reducing the dimension of complex signals (Santello et al., 1998;

Todorov and Ghahramani, 2004; Hug et al., 2011; Pale et al., 2020). However, when the method is used to only reduce the dimensionality of EMG signals, which are then utilized to control a separate system (*e.g.*, virtual hand), the importance of reconstruction diminishes altogether since EMG signals do not need to be reconstructed. What becomes important is the structure of the latent space representative of the complex myoelectric signals and its ability to match the hand kinematics latent space.

Here, we outline the factors that make a latent space useful in a control setting: class separability, latent dimension variance, and location of resting position.

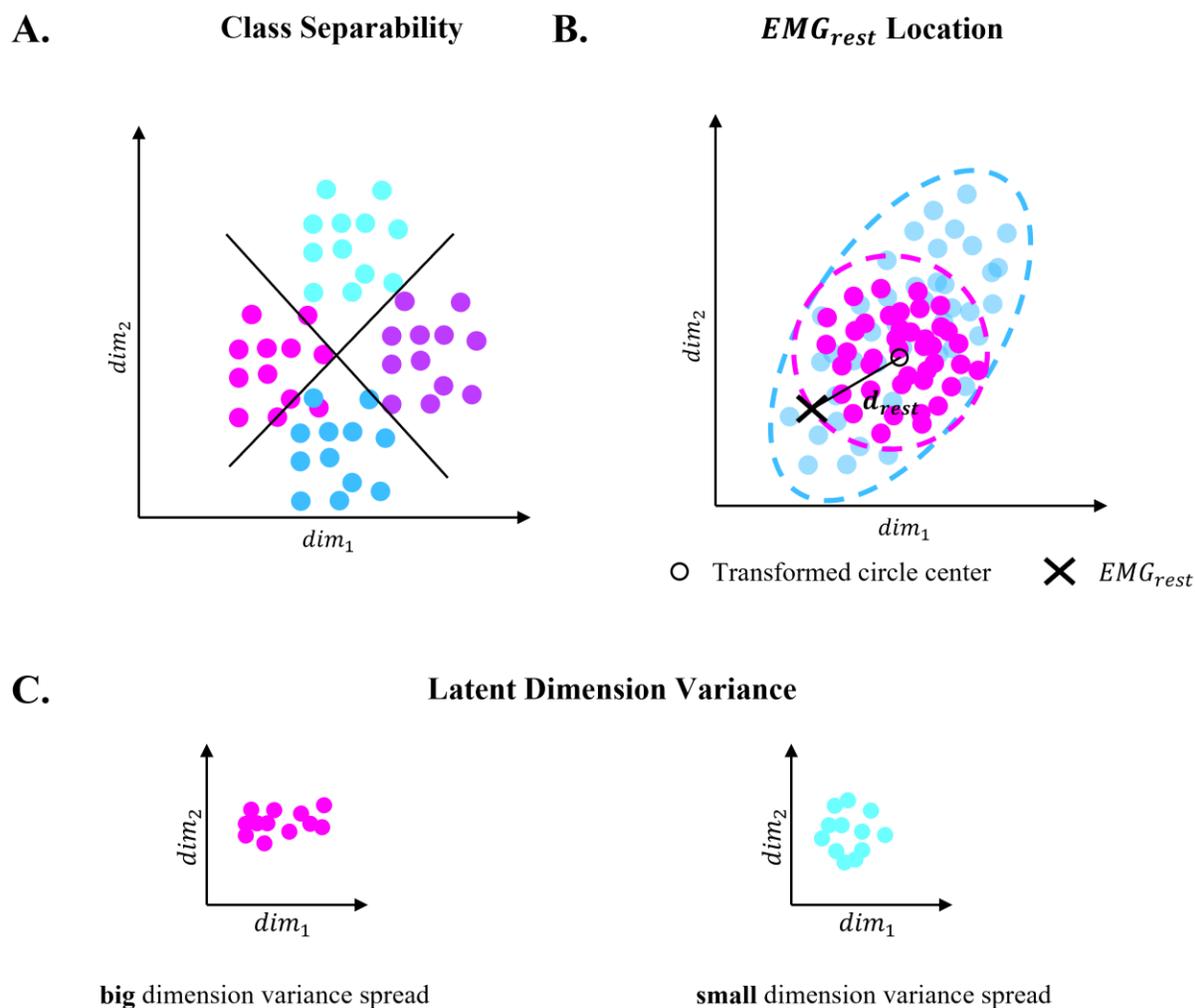


Figure 6-2. Visualization of latent space factors. **(A)** Class separability defined as a measure of how separable points of different classes on a latent space are with a straight line. **(B)** Location of the resting muscle signals, EMG_{rest} , encoded on a latent space. EMG_{rest} location is found by first finding the minimum-area ellipse that includes all points on the latent space (blue). The ellipse is then transformed to a unit circle (magenta). The location of EMG_{rest} is calculated by finding the Euclidian norm between the transformed EMG_{rest} and the center of the unit circle. **(C)** Latent dimension variance is defined how much variance each of the dimension has with respect to the sum of variances of both dimensions. Latent dimension variance spread is defined as the difference between the dimensions of maximum and minimum variances.

Class Separability

When designing a controller, creating a space where different movements are easily separable can be of great importance. By navigating along a more separable control space, the user might acquire

the ability to switch between different grasps and movements much faster than in cases where tasks are not separable.

Class separability was defined as how distinct the grasp/movement classes are when encoded onto the latent space (*Figure 6-2A*). It was calculated using SoftMax regression (Gao and Pavel, 2017). SoftMax regression was chosen as a simple example of a linear classification algorithm that does not require hyperparameter tuning. Higher accuracy percentage indicated a latent manifold in which classes (*i.e.*, grasps or different wrist movements) were more linearly separable.

SoftMax regression was applied to the 2D latent manifolds of the analyzed datasets. We used a five-fold cross-validation to calculate the average separability on each dataset.

Muscle Resting (EMG_{rest}) Location

In the following cases, we considered a myoelectric control that was based on the *position paradigm*, which meant that EMG activity was directly related to the position of the cursor on a 2D plane. Consequently, no EMG activity meant returning of the cursor to the original position. To sustain a given position, besides that of rest, the user must constantly contract muscles at a certain level which keeps the cursor at the desired location on the 2D plane.

As a result of such control, the position of EMG_{rest} (where no muscle contraction takes place) on the latent space becomes of high importance as it indicates where the cursor will return to when the muscles are at complete rest. Choosing a latent space with an EMG_{rest} position in the middle of all possible points encoded on the space makes it for an easier controller – the user is able to reach in different directions and can always come back to the center of the workspace, indicated as the neutral position. In cases of postural control, it can be defined as an open hand.

The visual representation of how we quantified the quality of the location of EMG_{rest} can be seen in *Figure 6-2B*. The step-by-step algorithm is described below:

First, we found an ellipse of minimum area that contained all points on the latent space (Moshtagh, 2005). The ellipse was then transformed into a circle, with the latent points scaled accordingly. To do so, we considered the minimum-area ellipse in its matrix form (*Equation 8*), where x is the point coordinate vector and Q is the 2×2 ellipse matrix.

$$x.' * Q * x = 1 \quad (8)$$

We then transformed the obtained ellipse to a unit circle using *Equation 9*, where x' is the transformed locations of the latent space, x is the original latent points, D is the diagonal matrix of Q , R is the orthogonal matrix of Q , and $c_{ellipse}$ is the location of the center of the obtained ellipse.

$$x' = D^{\frac{1}{2}} R (x - c_{ellipse}) \quad (9)$$

The transformed points were located on a unit circle. We then averaged the encoded resting positions to obtain a single point, representing EMG_{rest} , and calculated the magnitude of the vector from the center of the transformed circle, c_{circle} , to EMG_{rest} (*Equation 10*).

$$d_{rest} = \sqrt{(c_{circle,x} - EMG_{rest,x})^2 + (c_{circle,y} - EMG_{rest,y})^2} \quad (10)$$

Values of d_{rest} that were closer to 0 were considered to signify that the encoded EMG_{rest} was closer to the center of the latent space and, thus, more optimal than values closer to 1.

Latent Dimension Variance Spread

In cases of a lower-dimensional control, the variance associated with each latent dimension can be of great importance. By having equal distribution of variance across latent dimensions, one obtains a uniform controller in which motion is spread evenly across DOFs. One may argue that uneven variance distributions on latent spaces can be counteracted by applying one-dimensional scaling to the latent space to achieve an equal balance between dimensions. However, if one latent dimension has a significantly smaller variance than the others, “stretching” the latent space along that dimension will result in amplifying, what is effectively, noise. Such controller would then be not comfortable to use.

As a result, we propose the third latent space factor to be the spread of dimension variance, or the variance associated with each dimension in the latent manifold of a dimensionality-reduction model (*Figure 6-2C*). To calculate dimension variance, the input data were reduced to two latent dimensions for every dataset. 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 11*).

$$D = eig(cov(X)) \quad (11)$$

Each PCA dimension, PC_i , where i is 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 12*).

$$PC_i = diag(D(:, i)) \quad (12)$$

For AE and VAE, latent dimensions (CUs) were calculated by passing the normalized data *X* through the encoder part of the network (*Equations 2, 3*). CU_i is a corresponding column of $layer_2$ (*Equation 13*).

$$CU_i = \text{layer}_2(:, i) \quad (13)$$

Latent dimensions (PCs or CUs) are represented by A , an $m \times n$ matrix, where m is the number of observations and n is the number of latent dimensions. The mean of each latent dimension was calculated (*Equation 14*).

$$\mu = \frac{1}{m} \sum_{i=1}^m A_i \quad (14)$$

Afterwards, the variance of each latent dimension, v_i , was calculated:

$$v_i = \frac{1}{m-1} \sum_{i=1}^m \|A_i - \mu\|^2 \quad (15)$$

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 16*).

$$v_{dim} = \frac{v_i}{\sum_{i=1}^n v_i} \quad (16)$$

Lastly, what we calculated was the spread of variance by determining the difference between the dimensions of highest and lowest variance (*Equation 17*).

$$v_{spread} = \max(v_{dim}) - \min(v_{dim}) \quad (17)$$

The v_{spread} values of desire for an even controller were close to 0%.

Latent Space Analysis

PCA, AE, and VAE are applied to the EMG datasets and the resulting latent spaces are plotted and assessed for their usability in myoelectric control. To avoid considering the effect of random seed

variability on the variability of the latent space in AEs and VAEs, we used a fixed seed for all network training. Random seeds were only used in model training during the exploration of the dependency of the latent-space factors on the initialization weights.

The three latent space factors described above were obtained for every latent space and various aspects of the input data structures, such as participant's impairment level and EMG signal density.

Impairment Level

We looked at the effect of impairments (such as partial arm amputation) on the structure of the developed latent space. The amputee dataset was compared to the unimpaired dataset. Latent spaces created by PCA, AE, and VAE were evaluated.

Signal Density

In this section, we explored how having EMG data acquired with a high-density interface affected the latent space created by PCA, AE, and VAE. The high- and low-density datasets were compared, with the participants performing wrist flexion, extension, abduction, and adduction.

Latent Space Stability

The terminal structure of neural networks, such as AEs and VAEs, has a great dependency on the initial weights and biases, determined by the *seed*. This means that every time one initializes differently, the trained model arrives to a different minimum, resulting in potentially different latent space structure. In such cases, PCA has the advantage of stability as it optimizes variance without depending on the seed parameter.

Here, we explored how stable the latent space factors and trends were through different model initializations. To do so, we trained AE and VAE models 25 times with a random seed on the amputee, high- and low-density datasets.

Muscle Signal Correlation

We studied the overall muscle signal correlation, EMG_{corr} in each dataset and assessed its effects on the latent space structure. To do so, we first took each of the datasets that we performed latent space analysis on and constructed a pairwise correlation matrix between each signal pair (*Equation 18*).

$$EMG_{corr} = corr(EMG) \quad (18)$$

The matrix was then used for the reconstruction of a heatmap of correlation of each muscle signal pair. Then, we calculated the mean across all possible pairs to obtain the average EMG_{corr} . We used the Wilcoxon Rank-Sum test (Wilcoxon, 1945) to determine statistical differences between datasets in their muscle signal correlation (Mann and Whitney, 1947).

Results

Impairment Level

PCA, AE, and VAE were applied to unimpaired and amputee datasets, in which the participants performed five different grasps in a *dynamic sit* condition (*Figure 6-3*). The initialization seed was fixed for both AE and VAE. The class separability factor was similar across different datasets and dimensionality-reduction methods. The latent spaces derived from the amputee dataset appeared to have the least optimal EMG_{rest} locations for all three dimensionality-reduction methods and exhibited a marked “V” shape. The spread of dimension variance factor varied greatly across the datasets and methods.

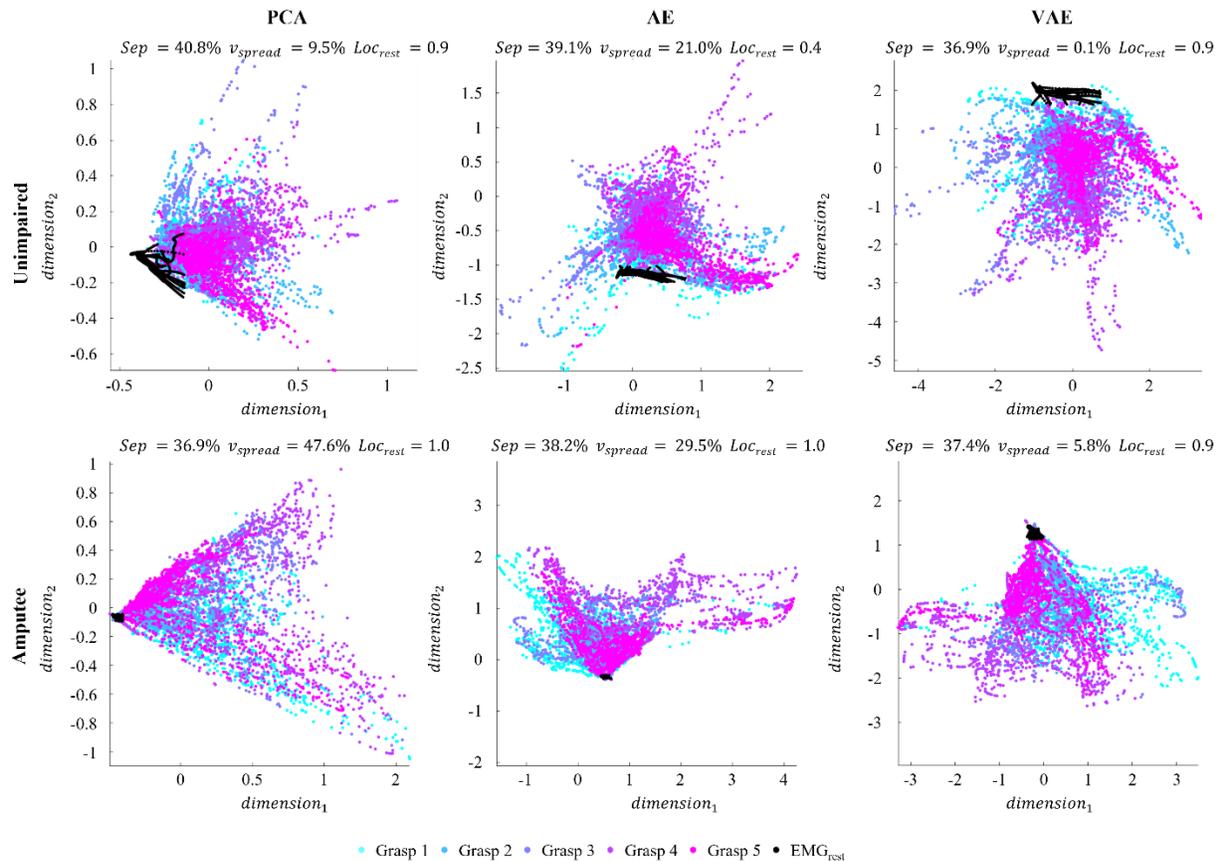


Figure 6-3. Visualization of 2D latent spaces produced by principal component analysis (PCA), autoencoder (AE) network, and variational autoencoder (VAE) network. Latent spaces are generated on the unimpaired (first row) and amputee (second row) datasets. The AE- and VAE-generated latent spaces are obtained with a fixed seed parameter. Grasps 1-5 are indicated by light blue, dark blue, purple, and pink colors, respectively. Both datasets were recorded in a *dynamic sit* condition, and the performed grasps were precision disk, prismatic pinch, index finger extension, adducted thumb, prismatic four-finger grasps. Black datapoints represent resting EMG signals encoded onto the latent space (EMG_{rest}). Sep is the class separability factor, v_{spread} is the spread of dimension variance, and Loc_{rest} is the location of the resting EMG, EMG_{rest} .

Let us now consider the factor stability plot for the datasets. In the unimpaired dataset, random weight initialization for both AE and VAE yielded a stable class separability of approximately 40% (Figure 6-4A). EMG_{rest} location had a large spread of possible values for both network types, and the spread was smaller in the VAE case. The spread of dimension variance for the AE network was big (from 0% to 75%), while it was small and stable for VAE.

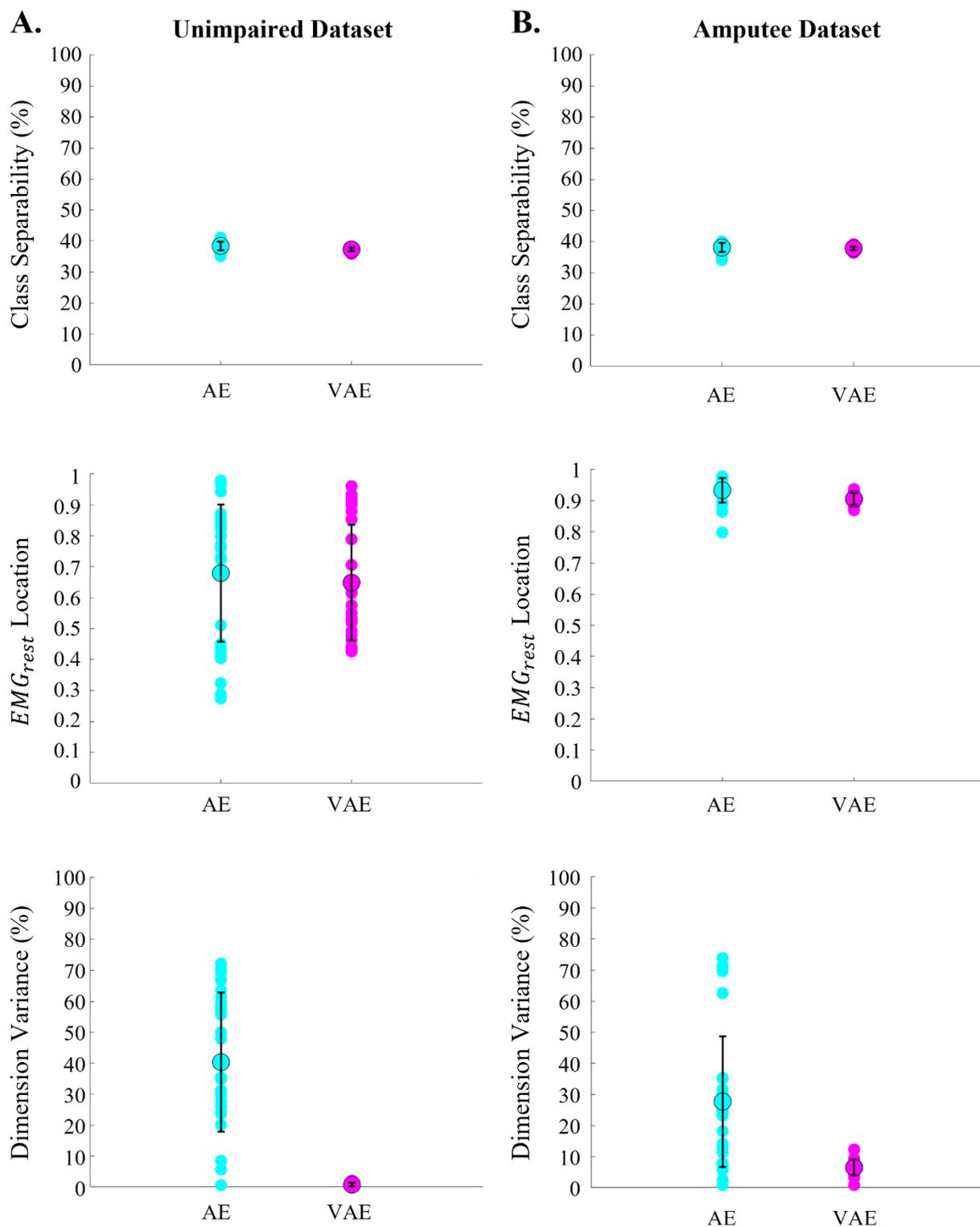


Figure 6-4. Stability of latent space factors (class separability, resting EMG, EMG_{rest} , location on the low-dimensional space, and spread of variance of two latent dimensions) for the unimpaired and amputee datasets. **(A)** Stability of class separability, EMG_{rest} location, dimension variance for unimpaired dataset considering random seed factor for autoencoder (AE) and variational autoencoder (VAE) networks. **(B)** Stability of class separability, EMG_{rest} location, dimension variance for the amputee dataset considering random seed factor for AE and VAE networks. Blue points indicate resulting factor values for the AE network. Magenta points indicate resulting factor values for the VAE network. Circles outlined in black indicate mean values while the errorbars signify standard deviation.

The amputee dataset yielded latent spaces with stable but less optimal EMG_{rest} location values (0.8 – 1.0 for AE and 0.85 – 0.95 for VAE) (*Figure 6-4B*). The class separability factor was also stable across random weight initializations, with relatively poor classification of approximately 38% for both neural networks. The dimension variance factor in AE had a large variability of approximately 80% while the VAE exhibited a significantly smaller variability of approximately 12%.

Signal Density

Latent space dependency on the density of the acquired signal was explored for the low- and high-density datasets (*Figure 6-5*). What changed dramatically between the datasets was the dimension of the input data (12 channels *vs* 256 channels). The most notable difference in the latent spaces was found in the spread of points for each class of wrist movement. The low-density dataset yielded latent spaces with a wider spread of points whereas the spread of points for the high-density dataset was narrower – almost lying on straight lines. It is important to note that these differences were not due to the difference in sample size between the datasets: prior to plotting, the low-density data were downsampled to match the size of the high-density data.

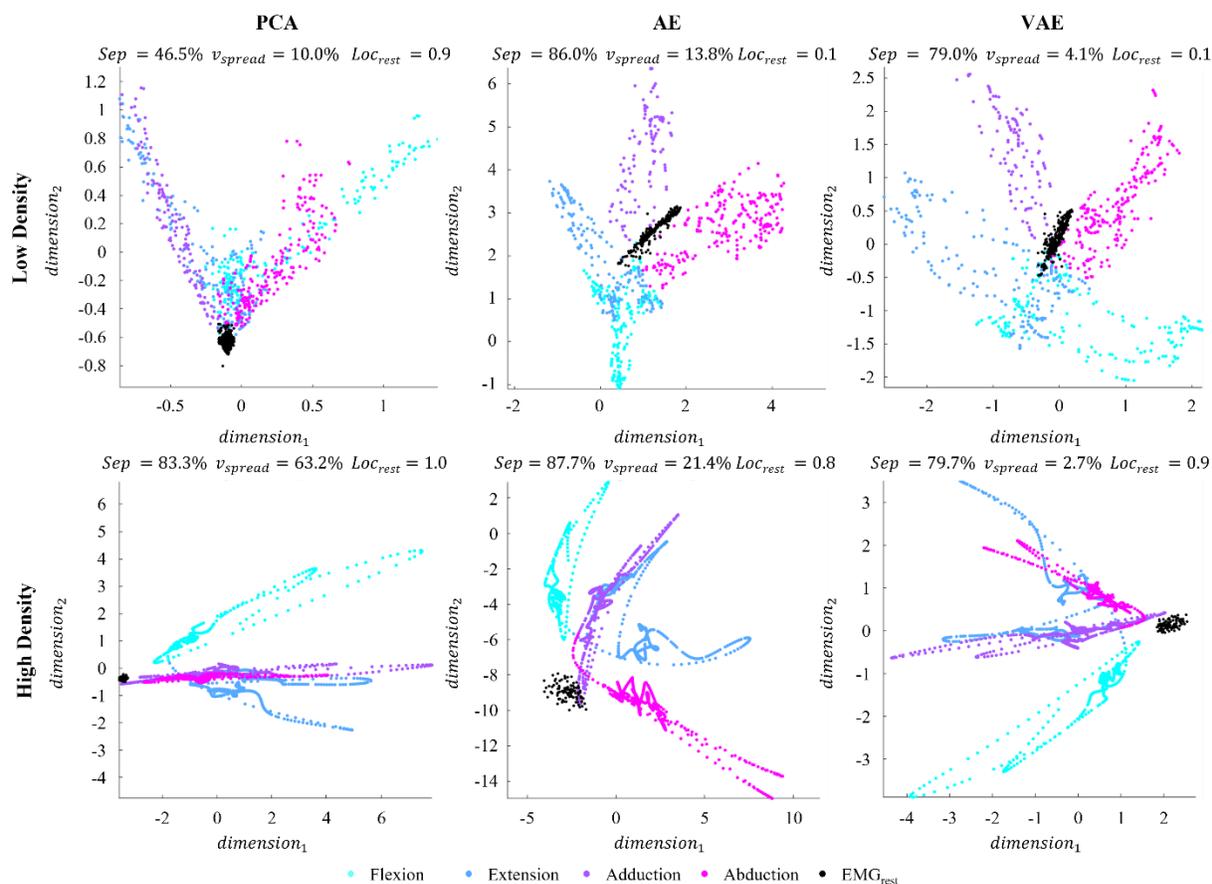


Figure 6-5. Visualization of the 2D latent spaces produced by principal component analysis (PCA), autoencoder (AE) network, and variational autoencoder (VAE) network. Latent spaces are generated on the low-density (first row) and high-density (second row) datasets. The AE- and VAE-generated latent spaces are obtained with a fixed seed parameter. Flexion, extension, adduction, and abduction of the wrist are indicated by light blue, dark blue, purple, and pink colors, respectively. Black datapoints represent resting EMG signals encoded onto the latent space (EMG_{rest}). Sep is the class separability factor, v_{spread} is the spread of dimension variance, and Loc_{rest} is the location of the resting EMG, EMG_{rest} .

In the high-density dataset, the EMG_{rest} point was at the bottom of the generated “V” shapes – similar to the PCA case of the low-density dataset. Such location yielded the least optimal EMG_{rest} values. The overall class separability for the high-density dataset was consistently high for the three dimensionality methods. In the case of low-density dataset, only the nonlinear networks produced latent spaces with high class separability while PCA did not.

Exploring the stability of the latent space factors in the high-density dataset, we first observe that class separability remained high regardless of random seed initialization, although VAE resulted in a significantly less separable latent space than AE (*Figure 6-6A*). In addition, the EMG_{rest} location factor was consistently poor for both nonlinear methods. In case of dimension variance spread, the trend was consistent with what was found in the amputee and unimpaired datasets: AE resulted in a large spread of variance, depending on the initialization seed, while VAE had a consistently stable outcome of variability less than 10%.

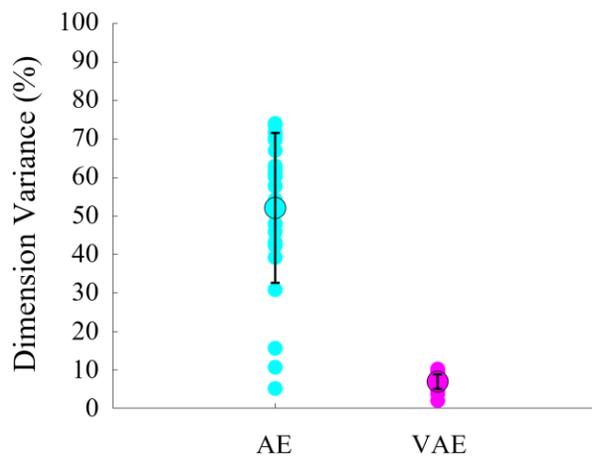
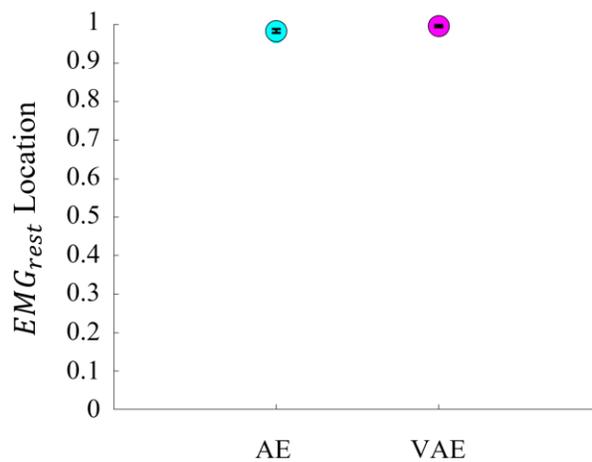
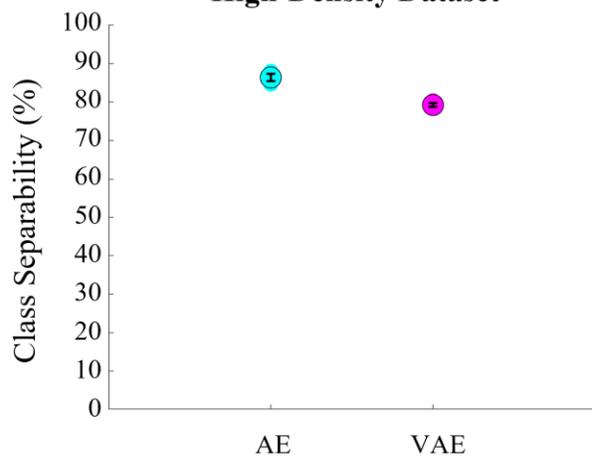
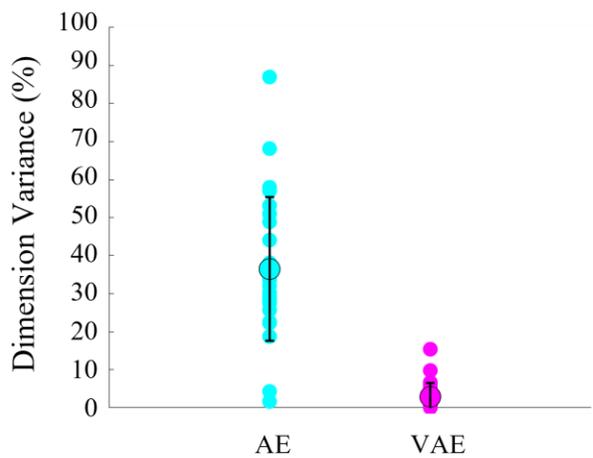
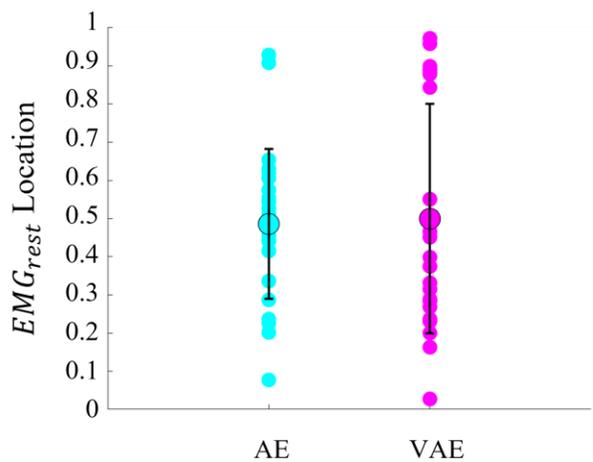
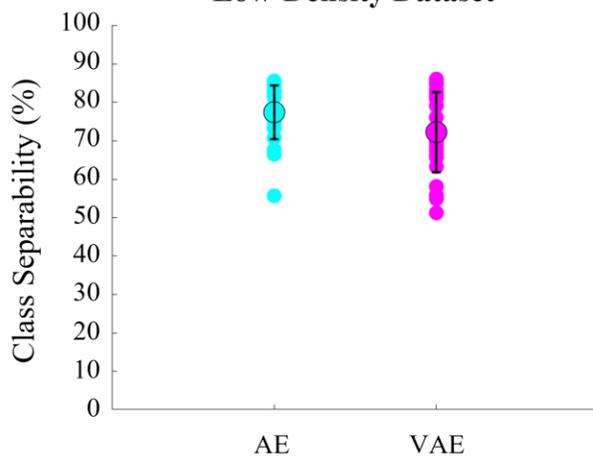
A. High-Density Dataset**B. Low-Density Dataset**

Figure 6-6. Stability of latent space factors (class separability, resting EMG, EMG_{rest} , location on the low-dimensional space, and spread of variance of two latent dimensions) for the high- and low-density datasets. **(A)** Stability of class separability, EMG_{rest} location, dimension variance for the high-density dataset considering random seed factor for autoencoder (AE) and variational autoencoder (VAE) networks. **(B)** Stability of class separability, EMG_{rest} location, dimension variance for the low-density dataset considering random seed factor for AE and VAE networks. Blue points indicate resulting factor values for the AE network. Magenta points indicate resulting factor values for the VAE network. Circles outlined in black indicate mean values while the errorbars signify standard deviation.

Latent-space factor stability was explored for the low-density data as well (*Figure 6-6B*). In case of AE, none of the factors were stable and produced varying results depending on the initialization parameter. Class separability, EMG_{rest} location, and dimension variance varied for 33%, 1.0, and 90%, respectively. Similar trend took place in the VAE network (35% and 1.0 variability in class separability and rest location, respectively). The only thing that remained consistent was the dimension variance spread, where the random seeds produced relatively stable outputs (within 11% of variability).

Muscle Signal Correlation

Here, we looked at how correlated the acquired muscle signals were for each of the datasets used in this study. By observing the heatmaps produced from the unimpaired datasets, one can see that, overall, EMG_{corr} was low (approximately 0.3) (*Figure 6-7A*). For the amputee dataset, the average EMG_{corr} increased significantly to 0.6, indicating higher correlation between collected muscle signals (*Figure 6-7B*). The low-density dataset resulted in a similar average EMG_{corr} as the acquisition conditions datasets (approximately 0.4) (*Figure 6-7C*). The high-density dataset, on the contrary, resulted in a higher index of correlation ($EMG_{corr} = 0.7$) (*Figure 6-7D*).

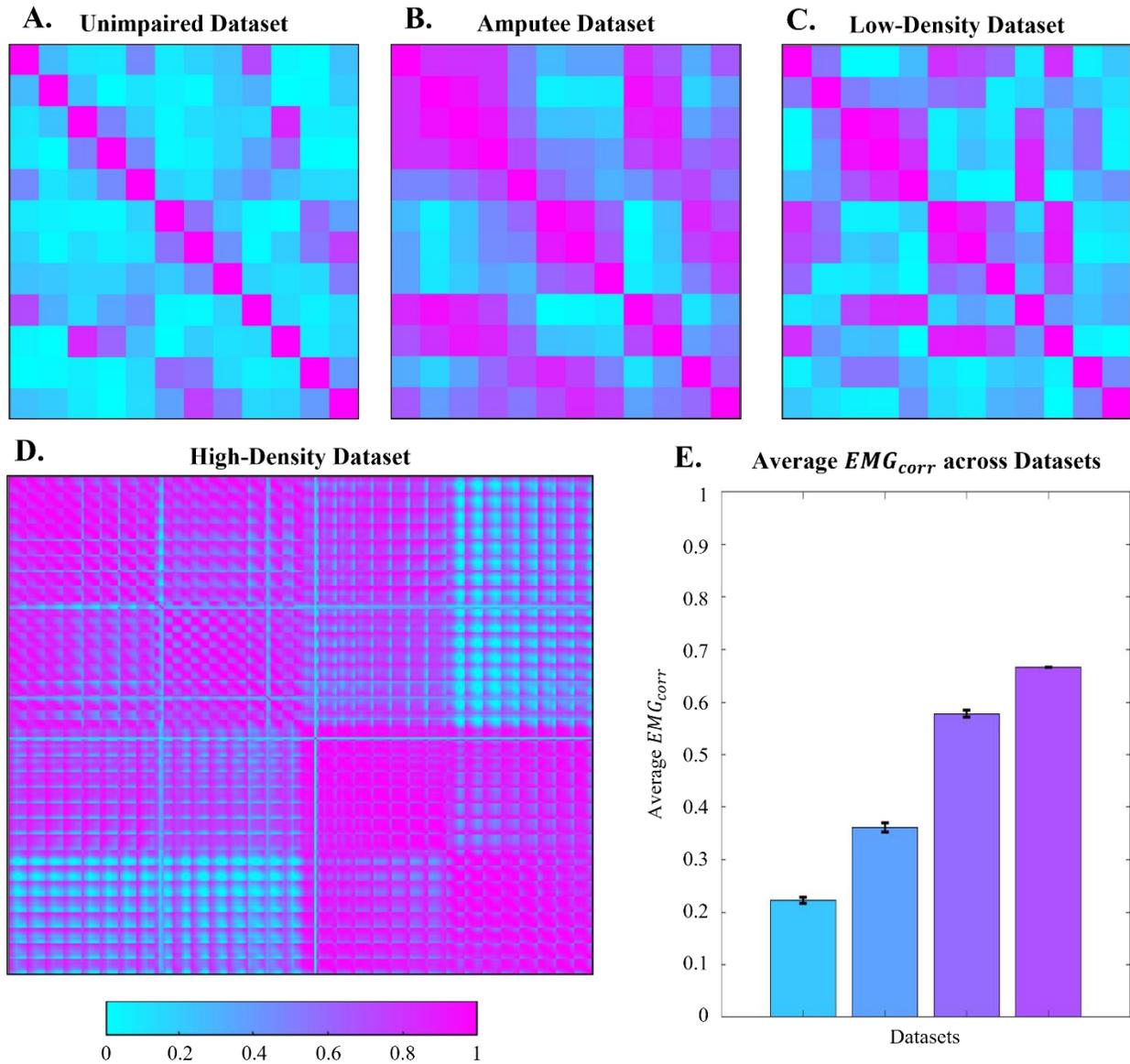


Figure 6-7. Correlation between muscle signals for different datasets. **(A)** Heatmaps of correlation between 12 acquired EMG signals for the *acquisition conditions dataset*. The four conditions are *dynamic sit*, *dynamic stand*, *static sit*, and *static stand*. **(B)** Heatmap of correlation between 12 acquired EMG signals for the *amputee dataset*. **(C)** Heatmap of correlation of 12 acquired EMG signals for the *low-density dataset*. **(D)** Heatmap of correlation of 256 EMG signals for the *high-density dataset*. **(E)** Average values EMG correlation (EMG_{corr}) for each of the dataset. The errorbars represent 95% confidence intervals. Double asterisk (**) indicates that the datasets yield EMG_{corr} values that are statistically different than any other dataset values. Single asterisk (*) indicates significant difference in EMG_{corr} values between the datasets.

Discussion

Structure of a latent space can be of high importance when considering low-dimensional controllers for high-dimensional systems. The way things are arranged when encoded has the potential to play a big role on the usability of the space for control purposes. As we discussed prior, latent spaces that have large spread in latent dimension variance yield unevenly distributed control across dimensions. Lacking high separability of classes in the low-dimension space can also result in the user's inability to switch between different classes in a fast and simple manner. Lastly, in a position controller, location of the resting EMG signals can play a big role in how much of the space can be covered by the user. In cases of poor EMG_{rest} locations, the myoelectric latent space results in the "V" shape mentioned throughout this paper. Such latent spaces cannot be mapped to the target kinematic latent space described in *Figure 6-1D* with a simple 1:1 transformation. As a result, we identified these factors as an important assessment tool to determine the appropriateness of the latent space for the purpose of low-dimensional myoelectric control.

In addition to proposing the factors, we assess the stability of latent spaces generated by neural networks with random initialization parameters. We determine the cases in which these factors remain stable and the cases that yield a big spread in the possible results. We also discuss the implications of the "V" shape that yields poorest EMG_{rest} location factor and determine the instances of input data that result in such a shape. Lastly, we outline the set of implications the results of our study have on the development of low-dimensional myoelectric controllers.

Latent Space Stability

Since PCA does not incorporate gradient descent, it has the advantage of resulting in a stable latent space, given the same input. However, when considering dimensionality-reduction methods, such as neural networks, structure of the latent space can be directly dependent on the initialization of

the weights and biases, thus resulting in a large variability in the latent-space factors evaluated in this study. What we discovered through this paper is that this effect can be strong for input data with certain characteristics while it does not appear in other data types.

Class Separability

The factor of class separability appeared to yield stable results for all four datasets (with the exception of a small variability in the low-density dataset) and for both AE and VAE networks. Class separability was low for the unimpaired and amputee datasets and high for the low- and high-density datasets. The main explanation for that can lie in the types of the movements that the participants performed in each dataset. The participants from the unimpaired and amputee datasets performed object grasps while the participants in the low- and high-density datasets performed simple wrist movements. It is possible that the differences in the class separability emerge due to the types of muscles involved in generating grasps or wrist movements. During wrist flexion, extension, abduction, and adduction, one is able to activate very specific groups of muscles that are “independent” from each other. However, in cases of object grasping, many different grasps, although completely unique from the kinematics perspective, may involve similar groups of muscles, which, in turn, resulted in less separable classes. As a result, this suggests that class separability was not really affected by the seed parameter, but mainly depended on the complexity of the encoded task.

Dimension Variance

AEs consistently yielded a large spread of possible dimension variance values across all datasets. VAEs, on the contrary, this latent space factor remained stable across random weight initializations. A possible explanation for such behavior may lie in the differences of the cost functions of AE and VAE, which every neural network aims to minimize. Since none of the latent-

space factors described in this paper are set to be optimized directly within the cost function of an AE network, model training does not result in stable outputs of these factors. Spread of latent dimension variance, however, is woven, although indirectly, into the VAE cost function, which, in turn, creates a stable output. We know that the KLD term in the VAE network aims to create a latent space that matches the normal Gaussian distribution. If achieved by the end of model training, a latent space of such shape would consequently result in an even distribution of variance over the two latent dimensions. As a result, this factor remains both stable and nearly optimal for the VAE model regardless of the seed. This trend of a stable variance spread factor in VAE networks is observed across all tested datasets.

Muscle Rest Location

The stability of the factor of EMG_{rest} location greatly depended on the type of the dataset that the neural network was applied to. Differences in the stability across different network types were insignificant, however. In cases of the unimpaired and the low-density datasets, both AE and VAE yielded a variety of possible EMG_{rest} locations – from more to least optimal values. On the contrary, when considering the high-density and amputee datasets, EMG_{rest} locations were stable and least optimal (near 1.0). The potential reason for this trend is discussed further in the next section.

The Curse of the V

When the encoded EMG_{rest} lies on or near the circumference of the transformed unit circle, the generated latent spaces take on the “V” shape mentioned throughout this paper. This shape was strongly prevalent in latent spaces created by PCA – we observed it in all datasets to which the linear dimensionality-reduction method was applied.

In cases of nonlinear neural networks, the “V” shape appeared in the high-density and amputee datasets, always resulting in the least optimal EMG_{rest} location values. In the cases of low-density and unimpaired datasets, however, it was not as prevalent and had a great dependency on initialization seed.

To understand what determines the prevalence of the “V” shape in AEs and VAES, we considered the correlation index for the collected EMG signals across different datasets. Datasets that consistently yielded the described shape and, consequently, poor EMG_{rest} locations, had high correlation indices ($EMG_{corr} > 0.6$). On the contrary, datasets that had a variability of shapes in the latent space had significantly lower EMG_{corr} values (between 0.3 and 0.4). At the moment, no clear explanation exists for this phenomenon and needs further exploration.

Implications

PCA vs Neural Networks

Although PCA exhibits stability in terms of the created structure of the latent space, they do, however, fall into the trap of yielding potentially unfavorable factors, such as unfavorable EMG_{rest} locations, consistently large spread of dimension variance (Portnova-Fahreva et al., 2020; Boe et al., 2021), and low class separability in cases of low-density data. Neural networks, such as AEs and VAES, on the contrary, can yield latent spaces with significantly more favorable factors, simply by changing the initialization seed. However, this only holds true for a specific input data structure, such as low-density datasets from unimpaired individuals.

AE vs VAE

One of the main advantages of VAE is their ability to generate small variance spread regardless of the initialization seed, unlike AE networks. This has the benefit of generating latent spaces with

more equally shared control between dimensions. In cases when the spread of dimension variance is large, the control of the dimension with the lowest variance may be equivalent to navigating along a noisy terrain.

In addition, VAEs are able to produce a potentially useful relationship between the factors of class separability and EMG_{rest} location, in which latent spaces with more optimal EMG_{rest} locations yield more separable classes. This can be useful when generating latent spaces used for myoelectric control purposes as we can assume that a more desirable shape can be achieved with a VAE, in which multiple factors are optimal. However, it is important to note once again that this relationship only holds true in a dataset with very specific characteristics.

Signal Density

High-density data, although advantageous in generating more separable classes in its latent space, also have the curse of generating the undesirable “V” shape. However, since this shape was only true for data with a high correlation index between signals, we hypothesize that having high-density data that is less correlated can potentially create latent spaces with more class separability and a more optimal EMG_{rest} location.

Users with Impairments

From this study, we can clearly observe the challenges that researchers may face when developing a low-dimensional myoelectric controller for amputees. Even with proper EMG placement, amputees appear to yield data with highly correlated signals – something that is not observed in their unimpaired counterparts during the same exercises. As a result, neither of the explored neural networks were able to produce latent spaces with desirable characteristics. One of the potential solutions to this might be to consider other dimensionality-reduction methods, outside of the

conventional AEs and VAEs, that would aid in creating latent spaces with more favorable factors for control purposes. In addition, EMG collected from body locations further away from the site affected by amputation may lead to potentially less correlated signals, which, in turn, would allow one to take advantage of the neural networks and their ability to generate more favorable latent spaces. However, in such cases, the naturalness of the developed controller may be lost as muscles that are not naturally involved in generating hand/wrist movements are used for control purposes.

Chapter 7 Concluding Remarks

Using other PCA-based controller methods that have been explored in the context of hand prosthetics as the basis of my research, I explored the applicability of nonlinear dimensionality-reduction techniques in the creation of a low-dimensional myoelectric controller. First of all, I have proved that nonlinear dimensionality-reduction methods, such as autoencoders, have a significant advantage over their linear counterparts when applied to high-dimensional biological signals (*i.e.*, hand kinematics, gait, and EMG). Their superiority is not only present in the reconstruction abilities but in the advantages of yielding a more balanced spread of variance across latent dimensions, which consequently can prove to be a useful asset in the development of the low-dimensional controller. In addition, the structural flexibility of neural networks allows them to effectively compress complex data and to control the practical shape of the encodings. Lastly, autoencoders have the advantage of including a temporal component, which allows for the incorporation of the time dynamics of the input data. As a result, these networks have a potential to be a powerful tool for hand prosthesis control.

In addition to developing an autoencoder-based controller, in which a high-dimensional hand system was controlled via a low-dimensional space, I found the factors that inhibit or aid learning of such controllers. These findings can serve as a platform for future studies of low-dimensional controllers and highlight 1) the need to separate between learning of the interface and learning of the controller and 2) the value of presenting a training task that explicitly indicates the connection between the underlying low-dimensional control space and the presented high-dimensional task space.

Lastly, I explored how dimensionality-reduction techniques can also be applied to the myoelectric end of the low-dimensional controller in order to match the latent spaces of EMG and hand kinematics. Through the last study, I developed a series of latent space factors (*i.e.*, class separability, dimension variance, and location of resting EMG) that one must consider when assessing the usefulness of the encoded EMG signals for control purposes. I discovered that while autoencoders have the advantage of having flexible structure, they can also yield unstable latent spaces (with different shapes and, hence, different factors) unlike PCA. Latent spaces derived from autoencoders greatly depend on the input data parameters, such as density signals and level of user's impairment, as well as the type of the neural network used for dimensionality reduction. Latent spaces of least optimal shapes result from data with high index of correlation between muscle signals. The findings of the last study can serve as a useful platform for using dimensionality-reduction methods on EMG signals for the purposes of low-dimensional control.

Future studies can explore if there is a difference in how people learn to operate linear and nonlinear low-dimensional controllers in the context of virtual hands. While through this PhD project, I have found a plethora of advantages that nonlinear neural networks, such as AEs, can have over their linear counterparts, PCA's rigidity and a set of rules that it abides to (*i.e.*, principles of orthogonality and superposition) have the potential to be advantageous when learning to operate low-dimensional controllers. Intuitively, these principles can aid the user in creating an internal map of the controller, consequently accelerating the learning process. On the contrary, AEs lack such principles, creating nonlinear latent space, that cannot be truly visualized with a simple transformation to a 2D plane. Nevertheless, it is important to point out that while not including any principles of orthogonality or superposition in its most basic forms, autoencoders are limitless

when it comes to their cost functions that they aim to optimize through training. As a result, it is possible to create neural networks that would, consequently, incorporate a stricter structure on the latent space; however, the performance of such networks can potentially be significantly different from what I have observed through my research.

References

- Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., et al. (Year). "{TensorFlow}: A System for {Large-Scale} Machine Learning", in: *12th USENIX symposium on operating systems design and implementation (OSDI 16)*, 265-283.
- Abdollahi, F., Farshchiansadegh, A., Pierella, C., Seáñez-González, I., Thorp, E., Lee, M.-H., 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.
- 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.
- Atzori, M., Gijsberts, A., Castellini, C., Caputo, B., Hager, A.-G.M., Elsig, S., et al. (2014). Electromyography data for non-invasive naturally-controlled robotic hand prostheses. *Scientific data* 1(1), 1-13.
- Azocar, A.F., Mooney, L.M., Duval, J.-F., Simon, A.M., Hargrove, L.J., and Rouse, E.J. (2020). Design and clinical implementation of an open-source bionic leg. *Nature biomedical engineering* 4(10), 941-953.
- Belter, J.T., Segil, J.L., and SM, B. (2013). Mechanical design and performance specifications of anthropomorphic prosthetic hands: a review. *Journal of rehabilitation research and development* 50(5), 599.
- Benatti, S., Milosevic, B., Farella, E., Gruppioni, E., and Benini, L. (2017). A prosthetic hand body area controller based on efficient pattern recognition control strategies. *Sensors* 17(4), 869.
- Beyeler, M., Rounds, E.L., Carlson, K.D., Dutt, N., and Krichmar, J.L. (2019). Neural correlates of sparse coding and dimensionality reduction. *PLoS computational biology* 15(6), e1006908.
- Biddiss, E., and Chau, T. (2007). Upper-limb prosthetics: critical factors in device abandonment. *American journal of physical medicine & rehabilitation* 86(12), 977-987.
- Boe, D., Portnova-Fahreeva, A.A., Sharma, A., Rai, V., Sie, A., Preechayasomboon, P., et al. (2021). Dimensionality Reduction of Human Gait for Prosthetic Control. *Frontiers in Bioengineering and Biotechnology*, 925.
- Bonferroni, C. (1936). Teoria statistica delle classi e calcolo delle probabilita. *Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commerciali di Firenze* 8, 3-62.
- Carrozza, M.C., Massa, B., Micera, S., Lazzarini, R., Zecca, M.e., and Dario, P. (2002). The development of a novel prosthetic hand-ongoing research and preliminary results. *IEEE/Asme Transactions on Mechatronics* 7(2), 108-114.

- Castellini, C. (2020). "Upper limb active prosthetic systems—overview," in *Wearable Robotics*. Elsevier), 365-376.
- Castellini, C., Artemiadis, P., Wininger, M., Ajoudani, A., Alimusaj, M., Bicchi, A., et al. (2014). Proceedings of the first workshop on peripheral machine interfaces: Going beyond traditional surface electromyography. *Frontiers in neurorobotics* 8, 22.
- Chang, C.-C., and Lin, C.-J. (2011). LIBSVM: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)* 2(3), 1-27.
- Chen, P.-H., Lien, C.-W., Wu, W.-C., Lee, L.-S., and Shaw, J.-S. (2020). Gait-based machine learning for classifying patients with different types of mild cognitive impairment. *Journal of medical systems* 44(6), 1-6.
- Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., et al. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.
- Ciocarlie, M.T., and Allen, P.K. (2009). Hand posture subspaces for dexterous robotic grasping. *The International Journal of Robotics Research* 28(7), 851-867.
- Cipriani, C., Zaccone, F., Micera, S., and Carrozza, M.C. (2008). On the shared control of an EMG-controlled prosthetic hand: analysis of user–prosthesis interaction. *IEEE Transactions on Robotics* 24(1), 170-184.
- Cognolato, M., Gijssberts, A., Gregori, V., Saetta, G., Giacomino, K., Hager, A.-G.M., et al. (2020). Gaze, visual, myoelectric, and inertial data of grasps for intelligent prosthetics. *Scientific data* 7(1), 1-15.
- Cui, P.H., and Visell, Y. (Year). "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: IEEE*, 2529-2532.
- De Groot, F., Jonkers, I., and Duysens, J. (2014). Task constraints and minimization of muscle effort result in a small number of muscle synergies during gait. *Frontiers in computational neuroscience* 8, 115.
- Della Santina, C., Piazza, C., Gasparri, G.M., Bonilla, M., Catalano, M.G., Grioli, G., et al. (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.
- Deluzio, K.J., Wyss, U.P., Zee, B., Costigan, P.A., and Serbie, C. (1997). Principal component models of knee kinematics and kinetics: normal vs. pathological gait patterns. *Human Movement Science* 16(2-3), 201-217.
- Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

- Dyson, M., Barnes, J., and Nazarpour, K. (2018). Myoelectric control with abstract decoders. *Journal of neural engineering* 15(5), 056003.
- Dyson, M., Dupan, S., Jones, H., and Nazarpour, K. (2020). Learning, generalization, and scalability of abstract myoelectric Control. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 28(7), 1539-1547.
- 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.
- 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.
- Gaiser, I.N., Pylatiuk, C., Schulz, S., Kargov, A., Oberle, R., and Werner, T. (2009). The FLUIDHAND III: A multifunctional prosthetic hand. *JPO: Journal of Prosthetics and Orthotics* 21(2), 91-96.
- 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*.
- Geethanjali, P. (2016). Myoelectric control of prosthetic hands: state-of-the-art review. *Medical Devices (Auckland, NZ)* 9, 247.
- Geissinger, J.H., and Asbeck, A.T. (2020). Motion inference using sparse inertial sensors, self-supervised learning, and a new dataset of unscripted human motion. *Sensors* 20(21), 6330.
- Hackett, D., Pippine, J., Watson, A., Sullivan, C., and Pratt, G. (2013). An overview of the DARPA autonomous robotic manipulation (ARM) program. *Journal of the Robotics Society of Japan* 31(4), 326-329.
- Haesloop, D., and Holt, B.R. (Year). "A neural network structure for system identification", in: *1990 American Control Conference: IEEE*, 2460-2465.
- Hahne, J.M., Schweisfurth, M.A., Koppe, M., and Farina, D. (2018). Simultaneous control of multiple functions of bionic hand prostheses: Performance and robustness in end users. *Science Robotics* 3(19), eaat3630.
- Hargrove, L., Englehart, K., and Hudgins, B. (Year). "The effect of electrode displacements on pattern recognition based myoelectric control", in: *2006 International Conference of the IEEE Engineering in Medicine and Biology Society: IEEE*, 2203-2206.
- Herssens, N., Van Crielinge, T., Saeys, W., Truijen, S., Vereeck, L., Van Rompaey, V., et al. (2020). An investigation of the spatio-temporal parameters of gait and margins of stability throughout adulthood. *Journal of the Royal Society Interface* 17(166), 20200194.
- Highsmith, M.J., Kahle, J.T., Bongiorno, D.R., Sutton, B.S., Groer, S., and Kaufman, K.R. (2010). Safety, energy efficiency, and cost efficacy of the C-Leg for transfemoral amputees: a review of the literature. *Prosthetics and orthotics international* 34(4), 362-377.
- Hochreiter, S., and Schmidhuber, J. (1997). Long short-term memory. *Neural computation* 9(8), 1735-1780.

- Hossain, M.Z., Sohel, F., Shiratuddin, M.F., and Laga, H. (2019). A comprehensive survey of deep learning for image captioning. *ACM Computing Surveys (CSUR)* 51(6), 1-36.
- Hug, F., Turpin, N.A., Couturier, A., and Dorel, S. (2011). Consistency of muscle synergies during pedaling across different mechanical constraints. *Journal of neurophysiology* 106(1), 91-103.
- Iandolo, R., Marini, F., Semprini, M., Laffranchi, M., Mugnosso, M., Cherif, A., et al. (2019). Perspectives and challenges in robotic neurorehabilitation. *Applied Sciences* 9(15), 3183.
- Ingram, J.N., K rding, K.P., Howard, I.S., and Wolpert, D.M. (2008). The statistics of natural hand movements. *Experimental brain research* 188(2), 223-236.
- Iqbal, N.V., and Subramaniam, K. (2018). A review on upper-limb myoelectric prosthetic control. *IETE Journal of Research* 64(6), 740-752.
- Ison, M., and Artemiadis, P. (2015). Proportional myoelectric control of robots: muscle synergy development drives performance enhancement, retainment, and generalization. *IEEE Transactions on Robotics* 31(2), 259-268.
- Jiang, N., Englehart, K.B., and Parker, P.A. (2008). Extracting simultaneous and proportional neural control information for multiple-DOF prostheses from the surface electromyographic signal. *IEEE transactions on Biomedical Engineering* 56(4), 1070-1080.
- Jiang, X., Liu, X., Fan, J., Ye, X., Dai, C., Clancy, E.A., et al. (2021). Open access dataset, toolbox and benchmark processing results of high-density surface electromyogram recordings. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 29, 1035-1046.
- Junior, J.J.A.M., Freitas, M.L., Siqueira, H.V., Lazzaretti, A.E., Pichorim, S.F., and Stevan Jr, S.L. (2020). Feature selection and dimensionality reduction: An extensive comparison in hand gesture classification by sEMG in eight channels armband approach. *Biomedical Signal Processing and Control* 59, 101920.
- 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*.
- Kramer, M.A. (1991). Nonlinear principal component analysis using autoassociative neural networks. *AIChE journal* 37(2), 233-243.
- Kuiken, T.A., Miller, L.A., Turner, K., and Hargrove, L.J. (2016). A comparison of pattern recognition control and direct control of a multiple degree-of-freedom transradial prosthesis. *IEEE journal of translational engineering in health and medicine* 4, 1-8.
- Kullback, S., and Leibler, R.A. (1951). On information and sufficiency. *The annals of mathematical statistics* 22(1), 79-86.
- Kyberd, P.J., Holland, O.E., Chappell, P.H., Smith, S., Tregidgo, R., Bagwell, P.J., et al. (1995). MARCUS: A two degree of freedom hand prosthesis with hierarchical grip control. *IEEE Transactions on Rehabilitation Engineering* 3(1), 70-76.

- Lambert-Shirzad, N., and Van der Loos, H.M. (2017). On identifying kinematic and muscle synergies: a comparison of matrix factorization methods using experimental data from the healthy population. *Journal of neurophysiology* 117(1), 290-302.
- LeCun, Y., Touresky, D., Hinton, G., and Sejnowski, T. (Year). "A theoretical framework for back-propagation", in: *Proceedings of the 1988 connectionist models summer school*, 21-28.
- Leo, A., Handjaras, G., Bianchi, M., Marino, H., Gabiccini, M., Guidi, A., et al. (2016). A synergy-based hand control is encoded in human motor cortical areas. *Elife* 5, e13420.
- 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.
- 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.
- Lim, B., and Zohren, S. (2021). Time-series forecasting with deep learning: a survey. *Philosophical Transactions of the Royal Society A* 379(2194), 20200209.
- Liu, X., and Scheidt, R.A. (2008). Contributions of online visual feedback to the learning and generalization of novel finger coordination patterns. *Journal of neurophysiology* 99(5), 2546-2557.
- Magenes, G., Passaglia, F., and Secco, E.L. (Year). "A new approach of multi-dof prosthetic control", in: *2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society: IEEE*, 3443-3446.
- Malhotra, M., Rombokas, E., Theodorou, E., Todorov, E., and Matsuoka, Y. (Year). "Reduced dimensionality control for the act hand", in: *2012 IEEE International Conference on Robotics and Automation: IEEE*, 5117-5122.
- Mann, H.B., and Whitney, D.R. (1947). On a test of whether one of two random variables is stochastically larger than the other. *The annals of mathematical statistics*, 50-60.
- 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), 1-13.
- 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.
- Matsushima, A., Yoshida, K., Genno, H., and Ikeda, S.-i. (2017). Principal component analysis for ataxic gait using a triaxial accelerometer. *Journal of neuroengineering and rehabilitation* 14(1), 1-7.
- Mauch, H.A. (1968). STANCE CONTROL FOR ABOVE-KNEE ARTIFICIAL LEGS. *Bulletin of Prosthetics Research* 10, 61.

- Mika, S., Schölkopf, B., Smola, A., Müller, K.-R., Scholz, M., and Rätsch, G. (1998). Kernel PCA and de-noising in feature spaces. *Advances in neural information processing systems* 11.
- Moshtagh, N. (2005). Minimum volume enclosing ellipsoid. *Convex optimization* 111(January), 1-9.
- 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.
- 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.
- O'Neill, P., Morin, E.L., and Scott, R.N. (1994). Myoelectric signal characteristics from muscles in residual upper limbs. *IEEE Transactions on Rehabilitation Engineering* 2(4), 266-270.
- Østlie, K., Lesjø, I.M., Franklin, R.J., Garfelt, B., Skjeldal, O.H., and Magnus, P. (2012). Prosthesis use in adult acquired major upper-limb amputees: patterns of wear, prosthetic skills and the actual use of prostheses in activities of daily life. *Disability and Rehabilitation: Assistive Technology* 7(6), 479-493.
- 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.
- Pale, U., Atzori, M., Müller, H., and Scano, A. (2020). Variability of muscle synergies in hand grasps: Analysis of intra-and inter-session data. *Sensors* 20(15), 4297.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al. (2011). Scikit-learn: Machine learning in Python. *the Journal of machine Learning research* 12, 2825-2830.
- Pettee, M., Shimmin, C., Duhaime, D., and Vidrin, I. (2019). Beyond imitation: Generative and variational choreography via machine learning. *arXiv preprint arXiv:1907.05297*.
- Phinyomark, A., Petri, G., Ibáñez-Marcelo, E., Osis, S.T., and Ferber, R. (2018). Analysis of big data in gait biomechanics: Current trends and future directions. *Journal of medical and biological engineering* 38(2), 244-260.
- Pierella, C., De Luca, A., Tasso, E., Cervetto, F., Gamba, S., Losio, L., et al. (Year). "Changes in neuromuscular activity during motor training with a body-machine interface after spinal cord injury", in: *2017 International Conference on Rehabilitation Robotics (ICORR)*: IEEE), 1100-1105.
- Pistohl, T., Cipriani, C., Jackson, A., and Nazarpour, K. (2013). Abstract and proportional myoelectric control for multi-fingered hand prostheses. *Annals of biomedical engineering* 41(12), 2687-2698.

- 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.
- 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.
- Quintero, D., Villarreal, D.J., Lambert, D.J., Kapp, S., and Gregg, R.D. (2018). Continuous-phase control of a powered knee-ankle prosthesis: Amputee experiments across speeds and inclines. *IEEE Transactions on Robotics* 34(3), 686-701.
- Rabbi, M.F., Pizzolato, C., Lloyd, D.G., Carty, C.P., Devaprakash, D., and Diamond, L.E. (2020). Non-negative matrix factorisation is the most appropriate method for extraction of muscle synergies in walking and running. *Scientific reports* 10(1), 1-11.
- Radhakrishnan, S.M., Baker, S.N., and Jackson, A. (2008). Learning a novel myoelectric-controlled interface task. *Journal of neurophysiology* 100(4), 2397-2408.
- Rai, V., and Rombokas, E. (Year). "A framework for mode-free prosthetic control for unstructured terrains", in: *2019 IEEE 16th International Conference on Rehabilitation Robotics (ICORR)*: IEEE), 796-802.
- Rai, V., Sharma, A., Preechayasomboon, P., and Rombokas, E. (Year). "Coordinated Movement for Prosthesis Reference Trajectory Generation: Temporal Factors and Attention", in: *2020 8th IEEE RAS/EMBS International Conference for Biomedical Robotics and Biomechatronics (BioRob)*: IEEE), 939-945.
- Rehbaum, H., Jiang, N., Paredes, L., Amsuess, S., Graimann, B., and Farina, D. (Year). "Real time simultaneous and proportional control of multiple degrees of freedom from surface EMG: preliminary results on subjects with limb deficiency", in: *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*: IEEE), 1346-1349.
- Reiter, R. (1948). Eine neue elektrokunsthand. *Grenzgebiete der Medizin* 1(4), 133-135.
- Rizzoglio, F., Pierella, C., De Santis, D., Mussa-Ivaldi, F., and Casadio, M. (2020). A hybrid Body-Machine Interface integrating signals from muscles and motions. *Journal of Neural Engineering* 17(4), 046004.
- Romero, J., Feix, T., Kjellström, H., and Kragic, D. (Year). "Spatio-temporal modeling of grasping actions", in: *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*: IEEE), 2103-2108.
- Rumelhart, D.E., Hinton, G.E., and Williams, R.J. (1985). "Learning internal representations by error propagation". California Univ San Diego La Jolla Inst for Cognitive Science).
- Sanders, J.E., and Fatone, S. (2011). Residual limb volume change: systematic review of measurement and management. *Journal of rehabilitation research and development* 48(8), 949.
- Sanders, J.E., Rogers, E.L., and Abrahamson, D.C. (2007). Assessment of residual-limb volume change using bioimpedance. *Journal of rehabilitation research and development* 44(4), 525.

- Sanger, T.D. (1989). Optimal unsupervised learning in a single-layer linear feedforward neural network. *Neural networks* 2(6), 459-473.
- Santello, M., Baud-Bovy, G., and Jörntell, H. (2013). Neural bases of hand synergies. *Frontiers in computational neuroscience* 7, 23.
- Santello, M., Flanders, M., and Soechting, J.F. (1998). Postural hand synergies for tool use. *Journal of neuroscience* 18(23), 10105-10115.
- 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.
- Schuster, M., and Paliwal, K.K. (1997). Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing* 45(11), 2673-2681.
- Seáñez-González, I., Pierella, C., Farshchiansadegh, A., Thorp, E.B., Wang, X., Parrish, T., et al. (2016). Body-machine interfaces after spinal cord injury: rehabilitation and brain plasticity. *Brain sciences* 6(4), 61.
- Segil, J.L. (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. (2015). Novel postural control algorithm for control of multifunctional myoelectric prosthetic hands. *Journal of rehabilitation research and development* 52(4), 449.
- Segil, J.L., and Controzzi, M. (2014). Comparative study of state-of-the-art myoelectric controllers for multigrasp prosthetic hands. *Journal of rehabilitation research and development* 51(9), 1439.
- Segil, J.L., and Huddle, S.A. (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.
- Simon, A.M., Ingraham, K.A., Fey, N.P., Finucane, S.B., Lipschutz, R.D., Young, A.J., et al. (2014). Configuring a powered knee and ankle prosthesis for transfemoral amputees within five specific ambulation modes. *PloS one* 9(6), e99387.
- Slijepcevic, D., Zeppelzauer, M., Gorgas, A.-M., Schwab, C., Schüller, M., Baca, A., et al. (2017). Automatic classification of functional gait disorders. *IEEE journal of biomedical and health informatics* 22(5), 1653-1661.
- Soares, D.P., de Castro, M.P., Mendes, E.A., and Machado, L. (2016). Principal component analysis in ground reaction forces and center of pressure gait waveforms of people with transfemoral amputation. *Prosthetics and orthotics international* 40(6), 729-738.
- 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.

- Stone, M. (1974). Cross-validated choice and assessment of statistical predictions. *Journal of the royal statistical society: Series B (Methodological)* 36(2), 111-133.
- Summerside, E.M., Kram, R., and Ahmed, A.A. (2018). Contributions of metabolic and temporal costs to human gait selection. *Journal of The Royal Society Interface* 15(143), 20180197.
- Sutskever, I., Vinyals, O., and Le, Q.V. (2014). Sequence to sequence learning with neural networks. *Advances in neural information processing systems* 27.
- Tenenbaum, J.B., Silva, V.d., and Langford, J.C. (2000). A global geometric framework for nonlinear dimensionality reduction. *science* 290(5500), 2319-2323.
- Thorp, E.B., Abdollahi, F., Chen, D., Farshchiansadegh, A., Lee, M.-H., Pedersen, J.P., et al. (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. (2007). Dimensional reduction in sensorimotor systems: a framework for understanding muscle coordination of posture. *Progress in brain research* 165, 299-321.
- Ting, L.H., Chvatal, S.A., Safavynia, S.A., and Lucas McKay, J. (2012). Review and perspective: neuromechanical considerations for predicting muscle activation patterns for movement. *International journal for numerical methods in biomedical engineering* 28(10), 1003-1014.
- 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. (Year). "Analysis of the synergies underlying complex hand manipulation", in: *The 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society: IEEE*, 4637-4640.
- Tresch, M.C., Cheung, V.C., and d'Avella, A. (2006). Matrix factorization algorithms for the identification of muscle synergies: evaluation on simulated and experimental data sets. *Journal of neurophysiology* 95(4), 2199-2212.
- Tucker, M.R., Olivier, J., Pagel, A., Bleuler, H., Bouri, M., Lamercy, O., et al. (2015). Control strategies for active lower extremity prosthetics and orthotics: a review. *Journal of neuroengineering and rehabilitation* 12(1), 1-30.
- Vincent, P., Larochelle, H., Bengio, Y., and Manzagol, P.-A. (Year). "Extracting and composing robust features with denoising autoencoders", in: *Proceedings of the 25th international conference on Machine learning*, 1096-1103.
- 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*. Springer), 91-109.

- Weghe, M.V., Rogers, M., Weissert, M., and Matsuoka, Y. (Year). "The ACT hand: Design of the skeletal structure", in: *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004: IEEE*, 3375-3379.
- Weiss, E.J., and Flanders, M. (2004). Muscular and postural synergies of the human hand. *Journal of neurophysiology* 92(1), 523-535.
- Wilcoxon, F. (1945). "Individual comparisons by ranking methods. *Biom. Bull.*, 1, 80–83".).
- Wu, G., Siegler, S., Allard, P., Kirtley, C., Leardini, A., Rosenbaum, D., et al. (2002). ISB recommendation on definitions of joint coordinate system of various joints for the reporting of human joint motion—part I: ankle, hip, and spine. *Journal of biomechanics* 35(4), 543-548.
- Yan, Y., Goodman, J.M., Moore, D.D., Solla, S.A., and Bensmaia, S.J. (2020). Unexpected complexity of everyday manual behaviors. *Nature communications* 11(1), 1-8.
- 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.
- Young, A.J., Smith, L.H., Rouse, E.J., and Hargrove, L.J. (2012). Classification of simultaneous movements using surface EMG pattern recognition. *IEEE Transactions on Biomedical Engineering* 60(5), 1250-1258.
- Zhou, X., Tien, R.N., Ravikumar, S., and Chase, S.M. (2019). Distinct types of neural reorganization during long-term learning. *Journal of neurophysiology* 121(4), 1329-1341.
- 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.