Fall 1-31-2016

Comparison of muscle synergies elicited from transcranial meganetic stimulation (tms) and voluntary movements

Yifei Wei
New Jersey Institute of Technology

ABSTRACT

COMPARISON OF MUSCLE SYNERGIES ELICITED FROM TRANSCRANIAL MEGANETIC STIMULATION (TMS) AND VOLUNTARY MOVEMENTS

by
Yifei Wei

A key question in motor control is the redundancy of musculoskeletal elements involved. This problem refers to as the degree of freedom problem. The Muscle Synergy Hypothesis is one of the hypotheses that aim to resolve the problem which defines that a muscle synergy is a combination of a small set of muscles activated at different levels, serving as a building block that constructs motor behaviors. A recent study (Overduin et al. 2012) demonstrated that muscle synergies decomposed by Nonnegative Matrix Factorization (NMF) from EMG patterns evoked by intra-cortical microsimulation (ICMS) in the monkey remarkably matched ones observed in naturalistic reach-and-grasp behaviors. Another study (Ajiboye et al. 2009) showed that synergies elicited from a small number of hand postures can allow prediction of hand postures in general. Inspired by aforementioned studies, the aim of this study was to investigate whether Transcranial Magnetic Stimulation (TMS) can elicit muscle synergies matching ones observed in voluntary movements in healthy human subjects and whether these synergies can serve as frameworks to predict EMG patterns evoked by either TMS or voluntary movements.

Five healthy right-handed subjects participated in the study. 8 hand muscles were recorded to capture either TMS-evoked motor evoked potential (MEP) and electromyography (EMG) resulted from subjects' shaping American Sign Language (ASL) letters and numbers. NMF was utilized to extract synergies from both MEP and EMG data. We observed 5 or 6 synergies can capture 90% of variance of original and

matched synergies of two classes. The reconstructions of the original datasets (V_{TMS}: MEP data; V_{vol}: EMG data; V_{rand}: Random data as control) from synergies (H_{vol} synergies elicited from ASL tasks; H_{TMS} synergies elicited from TMS) was done by the nonnegative least-square algorithm, and Proportion of Variance Accounted for (PAV) served as a measure to quantify the quality of the estimation, giving results $H_{vol} \rightarrow V_{vol}$: 0.92 ± 0.02; $H_{TMS} \rightarrow V_{TMS}$: 0.94 ± 0.02; $H_{vol} \rightarrow V_{rand}$: 0.53 ± 0.03; $H_{TMS} \rightarrow V_{rand}$: 0.53 ± 0.07; $H_{TMS} \rightarrow V_{vol}$: 0.70 ± 0.06; $H_{vol} \rightarrow V_{TMS}$: 0.79 ± 0.06.

In conclusion, we argue that cortical components may involve in encoding synergies and we also demonstrate the possibility of synergies serving as frameworks in predicting and explaining human hand postures in general.

COMPARISON OF MUSCLE SYNERGIES ELICITED FROM TRANSCRANIAL MEGANETIC STIMULATION (TMS) AND VOLUNTARY MOVEMENTS

by
Yifei Wei

**A Thesis
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Biomedical Engineering**

Department of Biomedical Engineering

January 2016

Blank Page

APPROVAL PAGE

**COMPARISON OF MUSCLE SYNERGIES ELICITED FROM TRANSCRANIAL
MEGANETIC STIMULATION (TMS) AND VOLUNTARY MOVEMENTS**

Yifei Wei

Dr. Sergei Adamovich, Thesis Advisor	Date
Associate Professor of Biomedical Engineering, NJIT	

Dr. Mesut Sahin, Committee Member	Date
Professor of Biomedical Engineering, NJIT	

Dr. Max Roman, Committee Member	Date
Assistant Research Professor of Biomedical Engineering, NJIT	

BIOGRAPHICAL SKETCH

Author: Yifei Wei

Degree: Master of Science in Biomedical Engineering

Date: December, 2015

Undergraduate and Graduate Education:

- Master of Science in Biomedical Engineering (Neural Engineering Track),
  New Jersey Institute of Technology, Newark, NJ, 2015

- Bachelor of Science in Bioengineering (Cellular and Medical Product Track),
  The University of Pittsburgh, Pittsburgh, PA, 2011

- Material Science and Engineering (Polymer Science Track),
  Zhejiang University of Technology, Hangzhou, Zhejiang, P. R. China, 2008

Major: Biomedical Engineering

Presentations and Publications:

Yarossi, M., Wei, Y., Adamovich, S., Tunik., "Comparison of TMS elicited and Voluntary Synergies of the Human Hand," Annual Meeting of Society for Neuroscience, Chicago 2015

To people who cares or cared about me particularly my parents for giving birth to me, being with me, and supporting me with your unconditional love

ACKNOWLEDGMENT

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 The Degree of Freedom Problem

The liberation of hands marks the difference of humans from other mammals, and it is the premise of the development of tools that tremendously change the life style of human beings. Hand structure is a guarantee for human beings to perform intricate and complex tasks for its complexity that involves muscles, bones, nerves, blood vessels and tendons. Such complexity allows a person to perform tasks that requires a great movement range and adaptability. However, the large number of musculoskeletal elements involved features a high number of degree of freedom (DOF) which increases the difficulty for central nervous system to control the human hand (Bernstein, 1967), and the control may be more complex particularly when time course is considered. Such problem refers to as the biomechanical redundancy which implies that the DOF due to the participation of various joints, muscles in a motor task is more than necessary to execute movements. In order to resolve the problem of the redundancy of DOF, many hypotheses have been developed to explain how the nervous system can result in a particular solution out of a large number of possible solutions that can give rise to a mutual motor goal, including Muscle Synergy hypothesis, Equilibrium Point hypothesis and Uncontrolled Manifold hypothesis. In this study, synergy will be the "protagonist", and will be given more discussion in detail.

1.2 What are Synergies?

1.2.1 The Definition of a Synergy

A succinct definition of synergy was given in d'Avella et al. 2003 that "Muscle synergies-coherent activations, in space or time, of a group of muscles have been proposed as building blocks that could simplify the construction of motor behavior". A more elaborated definition can be found in Turvey et al. 2007, in which a synergy refers to "a collection of relatively independent degrees of freedom that behave as a single functional unit – meaning that the internal degrees of freedom take care of themselves, adjusting to their mutual fluctuations and to the punctuations of the external force field, and do so in a way that preserves the function integrity of the collection". Both definitions infer to the reduction of the number of controlled units for multiple muscles can be activated in a synergetic way under one motor command referring to a synergy, while such dimensionality reduction of control in the multi-element muscular system does not sacrifice the flexibility. Many types of synergy models were proposed. The mathematical form of the synergy model – Feedforward Time-Invariant Synergy Model in this study is given in Chapter 2.5.1.

1.2.2 The Origin of the Proposal of the Synergy Hypothesis

The results of early studies of the spinal cord in vertebral species such frog, rat and cat using techniques like microsimulation, cutaneous stimulation led to the hypothesis that motor primitives may be modules that can be combined to generate movements and postures. (Bizzi et al. 1991; Giszter et al. 1993; Tresch and Bizzi 1999) In the previously cited studies, force fields activated by the stimulation electrode implanted in the spinal

cord were found to group in different ways when moving the electrode to different locations. Mussa-Ivaldi et al. 1994 found the force vectors can be summed due to the co-stimulation of two sites in the spinal cord. Moreover, a subsequent study Loeb et al. 2000 discovered that the similarity between the simulated robust synergies defined as combination of muscle activation and actual synergies they found previously, and both classes can produce stable force field. All these results argue the possible existence of synergies. Synergy may be called Motor Primitive elsewhere. Modular control refers to the same meaning as well. However, in the field of rehabilitation, synergy has different concept as which in the field of motor control.

1.3 Transcranial Magnetic Stimulation (TMS)

1.3.1 A Brief Introduction to the Mechanism of TMS

Transcranial magnetic stimulation (TMS) (Barker et al. 1985) is a non-invasive brain stimulation method. During the stimulation, the coil is supposed to be put tangentially with respect to the scalp (Figure 1.3). The illustration to the mechanism of TMS is given in Figure 1.2 in which induced current in the brain is produced by the megnatic field is caused by the electrical influx that is perpendicular to the coil that is supposed to placed tangential to the scalp. The brief and high-voltage electric induced current over the primary motor cortex (M1) can corespondingly give rise to a brief and synchronous muscle response, which is defined a motor evoked potential (MEP). (Figure 1.1).
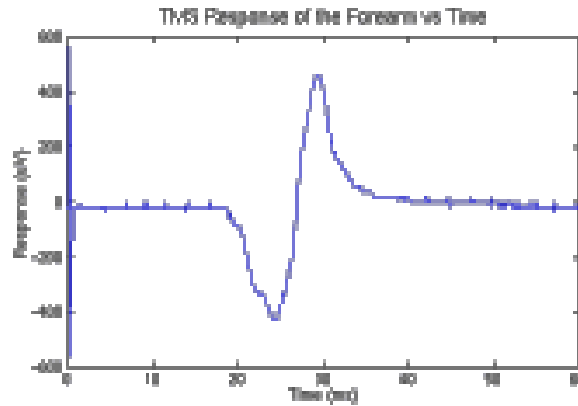
Figure 1.1 An illustration of MEP: TMS Response of the Forearms vs. Time.

Source:http://www.ai.mit.edu/projects/medical-vision/surgery/tms.html.
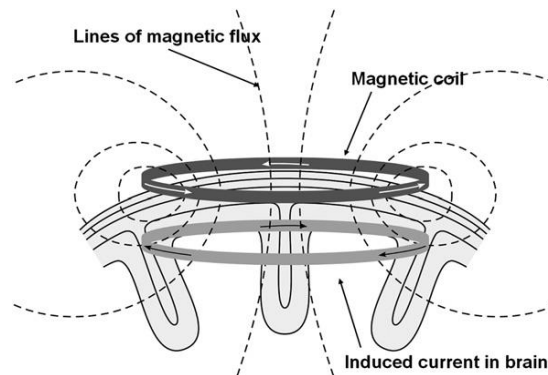
Figure 1.2 Stimulation mechnism of TMS. Current Flow direction in a Magnetic Coil and Induced Current in the Brain.

Source: Hallett et al. 2000.

1.3.2 Rationales of the Use of TMS in the Present Study

According to Rothwell et al. 1997, TMS can activate intracotical horizontal fibers when it is applied at a moderate intensity. These fibers are believed to be able to wire the corticospinal projection neurons to an extended cortical nework. (Huntley and Jones 1991). The primary motor cortex with its connected monosynaptic spinal neurons are believed to form anatomical units for finger movement. (Lang et al. 2004). TMS could stimulate

4

neuronal circuits and produce naturalistical movements or postures (Rothwell et al. 1997). In addition, Muscle Primitives are traditionally hypothesized to be encoded either in the spinal cord or brain stem (Bizzi et al. 2013). However, intraspinal microstimulation experiments in rats (Tresch and Bizzi 1999) demonstrated modular organization, experimental evidence for such functional modular organization was not observed in spinalized cats (Aoyagi et al. 2004; Mushahwar et al. 2004). We might argue that, if such modular organization exist, movements presented in higher-order animals than frog, i.e. cat, monkey, even humans, may require involvement of the primary motor cortex in realizing such modular control pattern. As a method that excites an extended region of the brain and of great convenience to use, TMS should be preferentially chosen, and we might argue the involvement of primary motor cortex in encoding synergies if TMS can elicit similar synergy patterns as voluntary movement does.
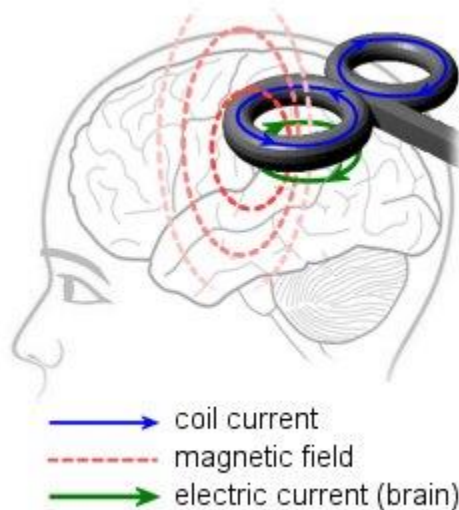
Figure 1.3 A demonstration of TMS.

Source: http://www.psychologypage.org/transcranial_magnetic_stimulation_tms.html.

5

1.4 Objective

The overal objective of this study is to explore the synergy hypothesis. In a recent study (Overduin et al. 2012), EMG patterns evoked by intracortical microstimulation (ICMS) in monkey subjects on areas of motor cortex connected to the spinal interneurons can be decomposed by NMF algorithm into a set of muscle synergies that mirror those obtained from reach-and-grasp behaviors. In an earlier study (Gentner and Classen, 2006), hand muscles' modular organization of cortical representations were suggested. In that study, PC analysis suggested that TMS used as a stimulation tool on primary motor cortex can elicit kinematic finger movements that were similar to those of grasping imagined objects by same subjects. Another study (Overduin et al. 2008) provided evidence that a small number of EMG synergies elicited from grasping objects could capture most variance in the original EMG dataset. In addition, Ajiboye et al. 2009 demonstrated that muscle synergies elicited from EMG recording of subjects performing American Sign Language (ASL) tasks can serves as not only a descriptive framework but also a predictive framework for a variety of hand postures. These studies inspired the form of the present study in which, other than using invsaive method, long duration ICMS, TMS, as a non-invasive brain stimulation method was applied for its convenience of use, harmlessness on human subjects and effectiveness in stimulation. EMG recording was favored instead of kinematic recording due to its convenience of implementation and the fact that EMG signals are reflections of net output from neural commands to alpha-motor neuron in the spinal which with the muscles form a motor unit. Unlike kinematic signals as an indirect evidence, EMG signals can be viewed as a more direct representation of muscle activation levels.

The specific aims of this study are to either investigate or confirm whether naturalistic activation of hand muscles can be evoked by TMS on human subjects; Whether TMS-evoked hand muscle responses exhibited modular properties; How many synergies decomposed by NMF algorithm (Explained in Chapter 2.5.2) are needed to capture a majority of variance in the original dataset; Whether TMS elicited synergies display a certain level of similarities with those resulted from voluntary hand movement; Whether those synergies demostrate a centain level of predictive power for either new sets of tasks or TMS elicited data sets, in other words that instead of being specific in terms of tasks, synergies are general and robust. The detailed experimental procedures, data analysis methods as well as representative results are given in the next chapter, and all other supplementary or intermediate results are provided in Chapter 3.

CHAPTER 2

METHODS AND RESULTS

In this chapter, all procedures including rationales behind will be explained in detail and typical results i.e. figures and tables will be provided for better illustration and explanation of the methods and results. MATLAB scripts used in data analysis are given in Appendices.

2.1 Subjects

Data were collected from five young ($\leq$ 40 years old) healthy right-handed subjects following informed consent.

2.2 EMG Channels

Eight right hand muscles were recorded by using EMG wireless electrodes during both TMS mapping session and American Sign Language (ASL) tasks session. (Figure 2.1) These muscles includes First Dorssal Interosseus (FDI), serving to abduct the index finger and rotate the index finger slightly at the metacarpophalangeal joint as well as assist adductor pollicis in thumb adduction; Extensor Indicus (EI), extending the index finger and assisting in wrist extension; Abductor Pollicis Brevis (APB), functioning in the abduction of the thumb; Abductor Digiti Minimi (ADM), playing an important role when grasping large objects with outspread fingers; Flexor Digitorum Superficialis (FDS), contributing to the flexion of the middle phalanges of the fingers at the proximal interphalangeal joints; Extensor Digitorum (ED), helping with the extension of the phalanges, wrist and the

elbow; Flexor Carpi Radials (FCR), assisting the flexion of the wrist and Extensor Carpi Radialis (ECR), finally, also extending the wrist. Electrode positions were verified by observing the EMG of corresponding muscle when asking the subject to perform a movement (i.e. FDI, pinch)
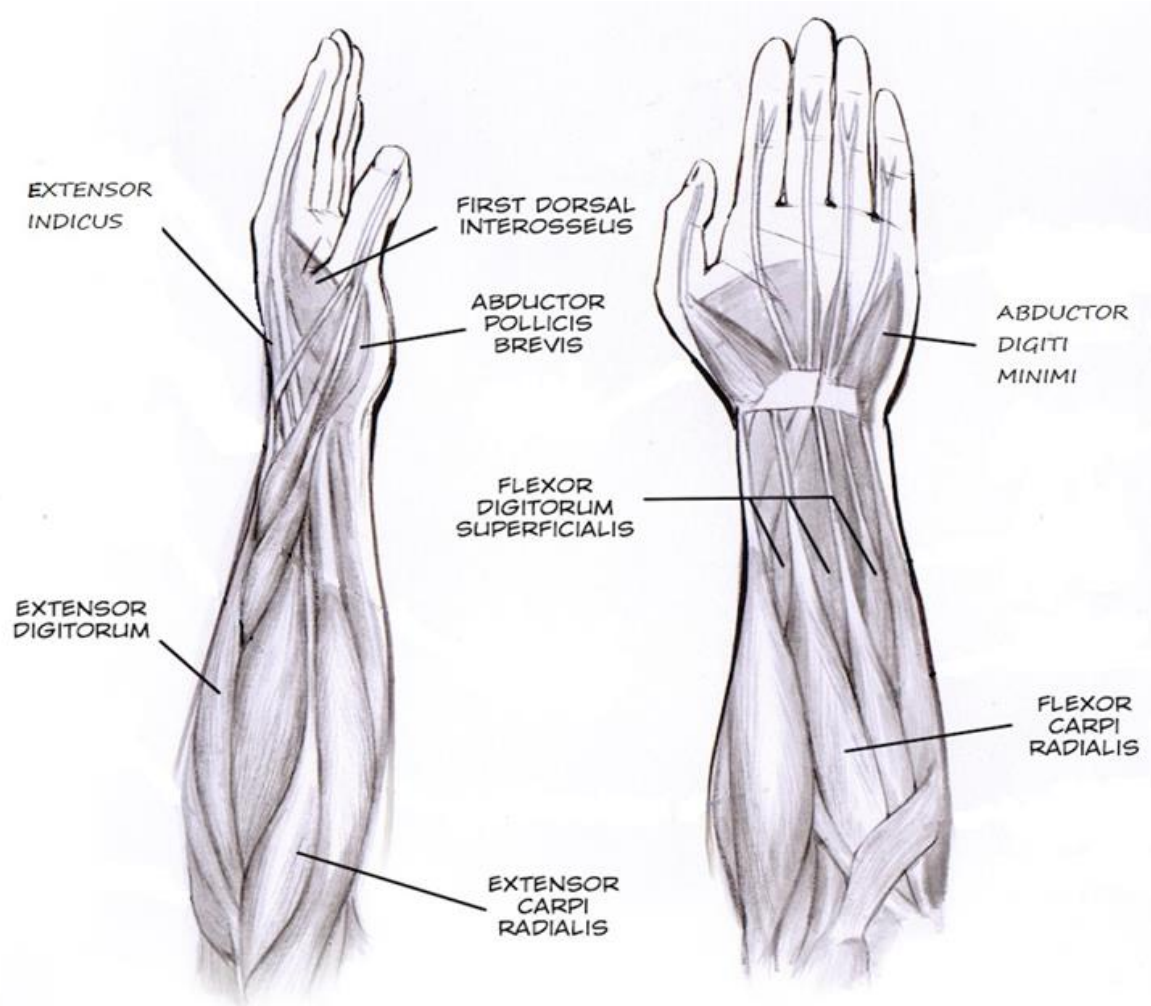
Figure 2.1 EMG recording channels. Eight right hand muscles were recorded including FDI, EI, APB, ADM, FDS, ED, FCR and ECR.

Source: Created by Mathew Yarossi.

2.3 TMS Mapping

2.3.1 Experiment Procedures

TMS mapping was conducted using a two-cone shaped magnetic coil. The coil is connected to a Magstim 200 monophastic stimulator (Magstim, Whitland, Dyfeld, UK). Motor evoked potentials (MEP) from aforementioned eight muscles were recorded. A neuronavigational workstation (Brainsight, Rogue Research, Montréal, Canada) recording the relative position of the coil with respect to the position of subject's head was used in the whole mapping process. The subject's head was registered with a template MRI scans. The optimal position for the magnetic coil to elicit MEPs in the right FDI muscle, termed as hotspot was found. It was defined as the position that TMS can produce the highest MEP on FDI at a moderate suprathreshold intensity on the motor cortex, where adjacent to the central gyrus. The minimal intensity of TMS on the hotspot that can produce response on the muscle is termed as the Resting Motor Threshold (RMT). The actual stimulations at a 110% RMT with an inter-stimulus interval (ISI) of 4 seconds were conducted on a 7cm x 7cm 10 dots x 10 dots stimulation grid with the FDI hotspot at the center in which each dot is a rough stimulation position. (Figure 2.2) On each dot on that grid, three stimulations were performed so that a total number of 300 stimulations were conducted for each subject. Stimulations not generate any MEP on any muscle were disregarded.
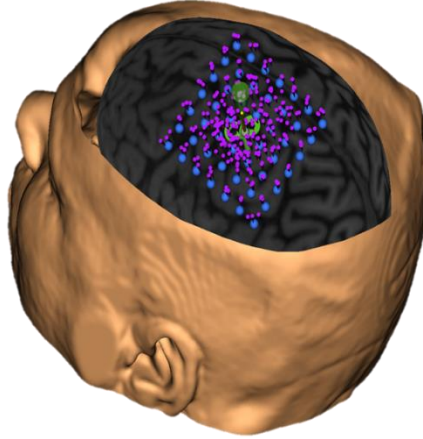
Figure 2.2 TMS schematic. Subject's head was registered with a template MRI scan. Green dot in the middle is the FDI hotspot. 10 x 10 blue dots on the 7cm x 7cm grid are instructing positions that serve to anchor the actual stimulations (Purple dots).

Source: Created by Mathew Yarossi.

2.3.2 MEP Data Preprocessing

The positions of stimulations were normalized and centered in analysis. The equations are given below:

$$Nomalized\ and\ Centered\ X\ position = \frac{X\ position - Maximal\ X}{Maximal\ X - Minimal\ X} \tag{2.1}$$

$$Nomalized\ and\ Centered\ Y\ position = \frac{Y\ position - Maximal\ Y}{Maximal\ Y - Minimal\ Y} \tag{2.2}$$

The MEP amplitudes for each channel are normalized to the largest MEP on that channel.

Amplitudes observed on that channel, and the matrix containing the normalized MEP amplitudes is given below:

$$\begin{bmatrix} N_{MEP_{1,1}} & \cdots & N_{MEP_{1,n}} \\ \vdots & \ddots & \vdots \\ N_{MEP_{m,1}} & \cdots & N_{MEP_{m,n}} \end{bmatrix}_{m\ x\ n}$$

Where m is the total channel number, and n is the total number of stimulations. Thus, each row is a eight dimensional vector, denoted as muscle vector or MV which stands for an aggregate of responses of 8 muscles resulted from a single stimulation. If no MEP for all channels was observed in a stimulation, that stimulation would be neglected and the corresponding row in the matrix would be removed from the matrix. The detailed MATLAB script for this procedure is attached in Appendix – MAP_Preprocess.m and MAP_Visualization.m for the visualization of the data. The TMS topographic results are given in the next section.

2.3.3 TMS Topographic Results Visualization
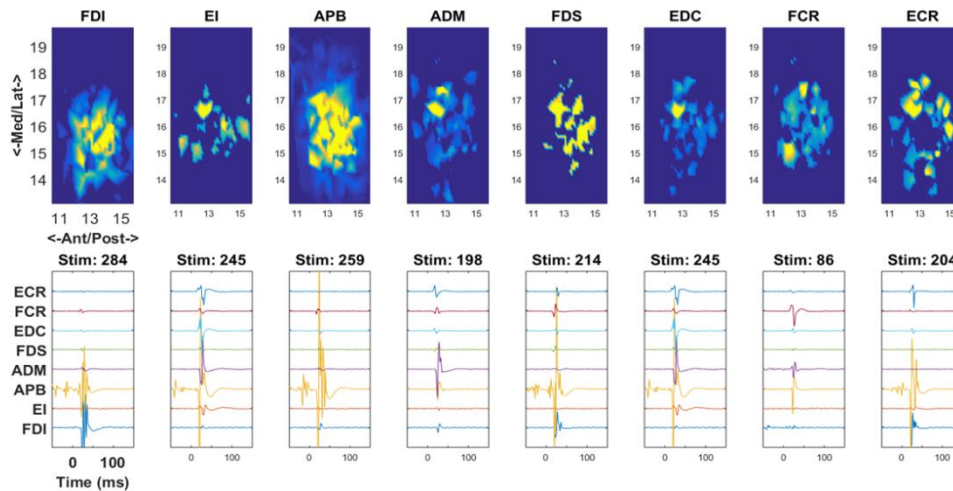
Figure 2.3 TMS results visualization. Muscle activation level vs TMS sites (Upper) and MEP for the corresponding stimulation sites (Lower). Subject S1.
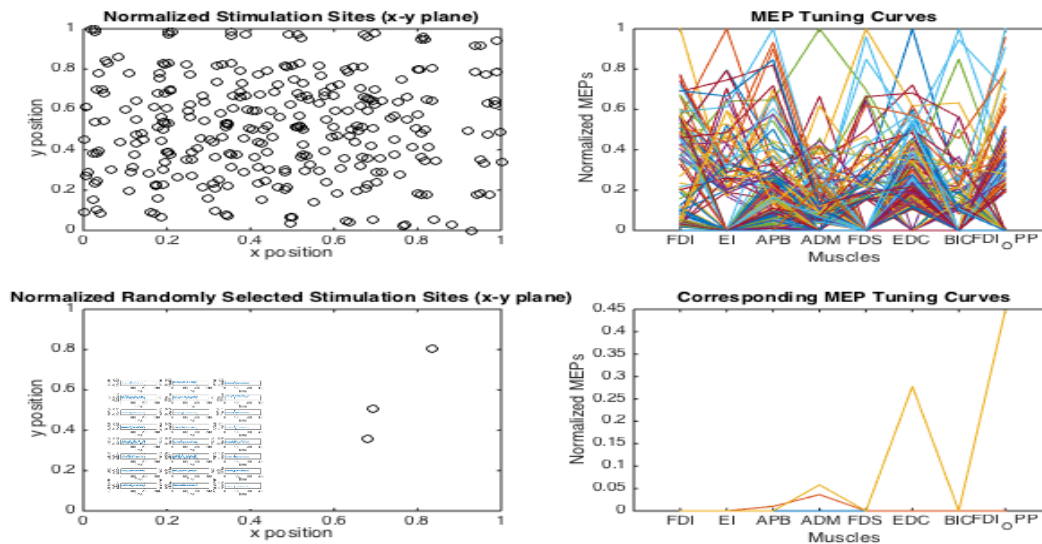
Figure 2.4 TMS representative results. (Upper-Left) Normalized Stimulation sites. (Upper-Right) Normalized MEP Tuning Curves for all stimulations. (Lower-Left) Three randomly selected stimulations. Embedded plot: MEP of the 3 randomly selected stimulations. (Lower-Right) Normalized MEP Tuning Curves of the 3 randomly selected stimulations. Subject S1.

2.4 American Sign Language Tasks

2.4.1 Experimental Procedure

Subjects were required to shape their hands into 33 static ASL letters in an ASL task trial for 5 to 7 seconds while watching and imitating a Youtube tutorial video at https://www.youtube.com/watch?v=9IEODEihHVw (Figure 2.5) (Subject S3 did not perform 6 through 9) Letters "J" and "Z" were omitted for they are not static. "0" was omitted because it is visually the same as "O". ("D" was neglected in the analysis for barely no EMG were observed for D) ASL tasks can provide sufficient spanning of hand postures thus when considering the total 32 shapes as a whole, all muscles were observed to be used and obvious responses were observed on EMG. Each subject was required to conduct three trials (In exception, Subject S3 performed only two trials). ASL sets are very similar to

hand postures when grasping objects. The reason that subjects were not required to familiarize themselves with the ASL sets beforehand is that natural postures were expected to be produced and the trace or influence of motor learning can be eliminated, and cohesion between trials would not be generated that EMG from all trials can be concatenated and considered a whole because EMG for a specific letter/number in different trials may not be exactly the same and can be considered different responses. Thus, the rank of the concatenated EMG matrix will not be lowered and can be fed in the NMF algorithm with no problems.

Figure 2.5 Static ASL letters and numbers. Subjects were instructed to shape their hands into 33 static ASL letters and numbers. Dynamic letters "J" and "Z" were omitted. "0" was omitted because it is visually the same as "O".

Source: Ajiboye et al. 2009.

2.4.2 EMG Data Preprocessing

EMG signals with a sampling frequency of 1000 Hz were filtered using a six-order Butterworth bandpass filter with cutoff frequencies of 10 Hz (Low) and 300 Hz (High). Another second-order Butterworth notch filter with cutoff frequencies of 59.5 Hz (Low) and 60.5 Hz (High) was applied for the purpose of removing noise resulted from power line at 60 Hz. Rectified and filtered EMG signals inside a time window from 5 second to 7 second were used for the remaining analysis (Figure 2.6), for stable value of EMG responses were observed during the time window. Thus, instead of integrating the EMG, averaged amplitude of the time window was used as a measure of activation level of muscle activities.
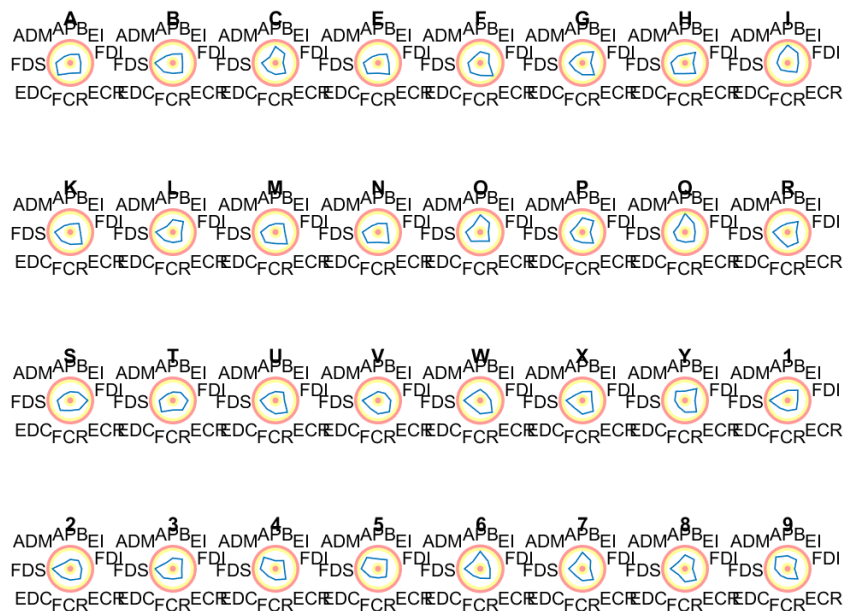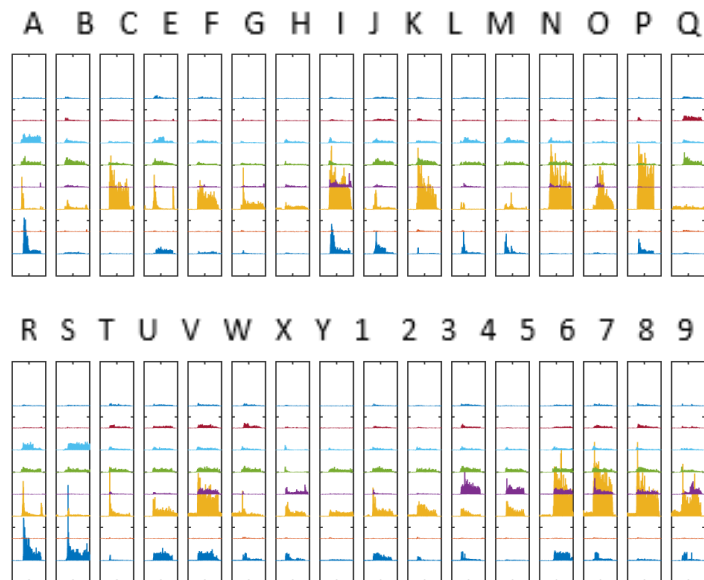
Figure 2.6 EMG patterns of ASL tasks. Upper: Rectified Raw EMG for all 32 ASL letters and numbers. Lower: (Subject S1 Trial 3) the outmost circle indicates value one for the normalized EMG amplitude.

Similar with how MEP matrix was constructed. The EMG amplitudes for each channel are normalized to the largest EMG amplitude observed on that channel within each trial.

$$\begin{bmatrix} N_{EMG\,1,1} & \cdots & N_{EMG\,1,n} \\ \vdots & \ddots & \vdots \\ N_{EMG\,m,1} & \cdots & N_{EMG\,m,n} \end{bmatrix}_{m\;x\;n}$$

Each row of the matrix represents EMG activity of each muscle when performing one sign. The detailed MATLAB script for this procedure is given in the Appendix – Vol_Mov_Preprocess.m

2.5 Muscle Synergies Extraction

The MATLAB source code can be found in the Appendix – synergy_extraction.m and synergy_extration_no_plot.m.

2.5.1 Time-Invariant Muscle Synergy Model

The concise generalized time-invariant feed-forward synergy model is illustrated in Figure 2.7.
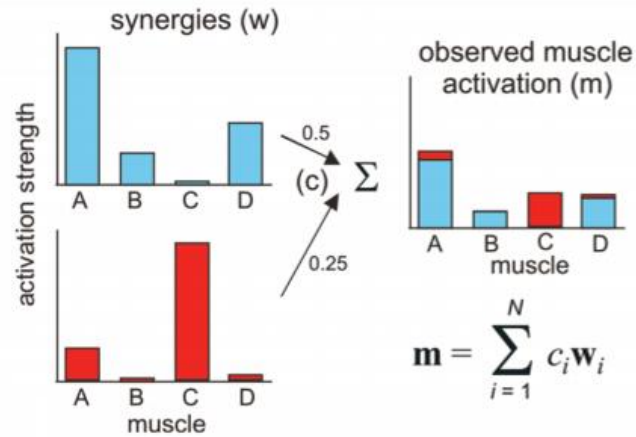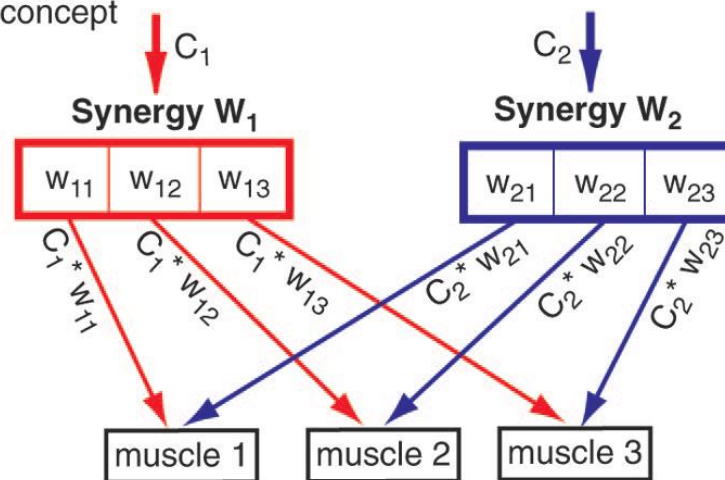
Figure 2.7 Schematic of Muscle Synergy Hypothesis. (Upper) The observed pattern (m) of multiple muscle (A, B, C, D) activities is a combination of two synergies (blue and red), each with a scaling factor c (0.5 for the blue synergy and 0.25 for the red one). (Lower)Three muscles are recruited in two synergies. Each element w_{ij} in a synergy vector W is the activation level of a given muscle j where i is the index of the synergy. A motor task that requires recruitment of these two synergies can be represented as an ensemble of scaling factor C multiply by each synergy vector W.

Source: Tresch et al.(Upper); Ting et al. 2005 (Lower).

A matrix form of the model is given as:

$$N_{m \times n} = W_{m \times k} \times H_{k \times n} \tag{2.3}$$

$$N_{m \times n} = \begin{bmatrix} N_{MEP} or N_{EMG\,1,1} & \cdots & N_{MEP} or N_{EMG\,1,n} \\ \vdots & \ddots & \vdots \\ N_{MEP} or N_{EMG\,m,1} & \cdots & N_{MEP} or N_{EMG\,m,n} \end{bmatrix}_{m \times n}$$

$$W_{m \times k} = \begin{bmatrix} W_{1,1} & \cdots & W_{1,k} \\ \vdots & \ddots & \vdots \\ W_{m,1} & \cdots & W_{m,k} \end{bmatrix}_{m \times k}$$

$$H_{k \times n} = \begin{bmatrix} H_{1,1} & \cdots & H_{1,n} \\ \vdots & \ddots & \vdots \\ H_{k,1} & \cdots & H_{k,n} \end{bmatrix}_{k \times n}$$

Where $N_{m \times n}$ (n muscles, m stimulations or ASLs denoted as observations) is either an MEP matrix or an ASL task related EMG matrix. Each row denotes as an observation resulted from either a TMS or an ASL; $H_{k \times n}$ is the synergy matrix contains k synergies. As given in the model, each row corresponds a synergy in which each element is the weight of corresponding muscle activation (i.e. $H_{2 \times 4}$: activation of muscle #4 in synergy #2); each column in the matrix $W_{m \times k}$ (k synergies) is weights of synergies of that corresponding observation (i.e. $W_{2 \times 4}$: weight of synergy #4 in observation #2). A hybrid use of NMF algorithms that is recommended by MATLAB documentation was implemented to decompose N and gave rise to H and W.

2.5.2 Muscle Synergy Extraction Algorithm

NMF is a versatile algorithm which has a lot of applications either in study of physiological problems. Image processing is another field where this algorithm has a wide use. In this study, two simplest and popularest NMF algorithms were used altogether to perform the decompossition that captures synergies: ALS and Mult. Unlike other algorithms that are

also frequently used in synergy study such as PCA, ICA, FA etc., the non-negative constraint for the NMF makes more physiological sense for the activations of muscles being positive make more sense. Other than that, as what was stated in Tresch et al. 2006 a paper reviews serveral decomposition methods used in the extraction of synergies, these algorithms "differ in their assumptions on two issues: on the distributions of activation coefficients and on the noise within the data set". For example, PCA and FA "assume a Gaussian distribution of activation coefficient", while NMF has "no explicit assumptions about the distributions of activation coefficient" which they believe results in the robust performance of NMF. That two commonly used NMF algorithms ALS and Mult have gained a large popularity is partially due to their simplicity and not requiring preset of algorithmic parameters while still producing reasonalbe results. The Matlab documentation states that "In general, the 'als' algorithm converges faster and more consistently, but was argued to have a problem of converging on local minima instead of global minimum. The 'mult' algorithm is more sensitive to initial values, which makes it a good choice when using 'replicates' to find W and H from multiple random starting values." Therefore, their recommended solution is the combined use of ALS and Mult algorithm as a hybrid method proposed by Berry et al. 2006. Below is my MATLAB source code sample conducting the synergy extraction procedure.

```
% A possible solution to find more consistent solutions
opt = statset('MaxIter',100,'Display','final');
% Maximal iteration allowed = 100; Display final outcome
[W0,H0] = nnmf(N_MEPamp,i,'replicates',10,'options',opt,'algorithm','mult');
opt = statset('Maxiter',10000,'Display','final');
[W,H,D] = nnmf(N_MEPamp,i,'w0',W0,'h0',H0,'options',opt,'algorithm','als');
```

In the MATLAB implementation, Mult determined the initial W_0 and H_0 which later needed to be fed in ALS algorithm for the final solution. H*W that has the lowest least

root-mean-square (RMS) among 10 attempts would be considered initial input $H_0 * W_0$ for the ALS algorithm. We visually confirmed that this hybrid method produced similar synergies by running the same dataset with this algorithms for many times.

2.5.3 The Determination of Synergy Number

≈90% of Proportion of Variance Account for (PAV) was set as a criterion when determining synergy numbers when applying NMF algorithm. The formula is given below:

Proportion of Variance Account for (PAV)

$$PVA = 1 - \frac{UV}{OV} \tag{2.4}$$

$$UV = \sum_{n=1}^{N} \sum_{i=1}^{L} (X_{ni} - E_{ni})^2 \tag{2.5}$$

$$OV = \sum_{n=1}^{N} \sum_{i=1}^{L} (X_{ni} - A)^2 \tag{2.6}$$

UV: Unexplained Variance

OV: Overall Variance

X_{ni}: Element in the original matrix N_EMG or N_MEP

E_{ni}: Estimated Matrix W*H

A: Expectation for each observation (Mean value of each observation): stimulation or ASL

In this process, NMF algorithm was repeatedly applied to decompose either MEP matrix or ASL task related matrix with different preset numbers of synergies from 1 to n, which is the dimensionality of MV, and is eight in our study for eight muscles were recorded. PAVs were calculated for every estimated matrix W*H of different synergy numbers. Whichever k generated the PAV closest to 90% was considered the number of

synergies and the corresponding H which was denoted as synergy matrix in the previous section was chosen for further studies. A typical illustration of this procedure is given in Figure 2.8, in which both the upper and lower subplot indicate that five synergies can explain approximately 90% of the variance of variability of either the MEP matrix obtained from Subject S1 or the ASL task related EMG matrix acquired from Subject S1 ASL task trial 3.
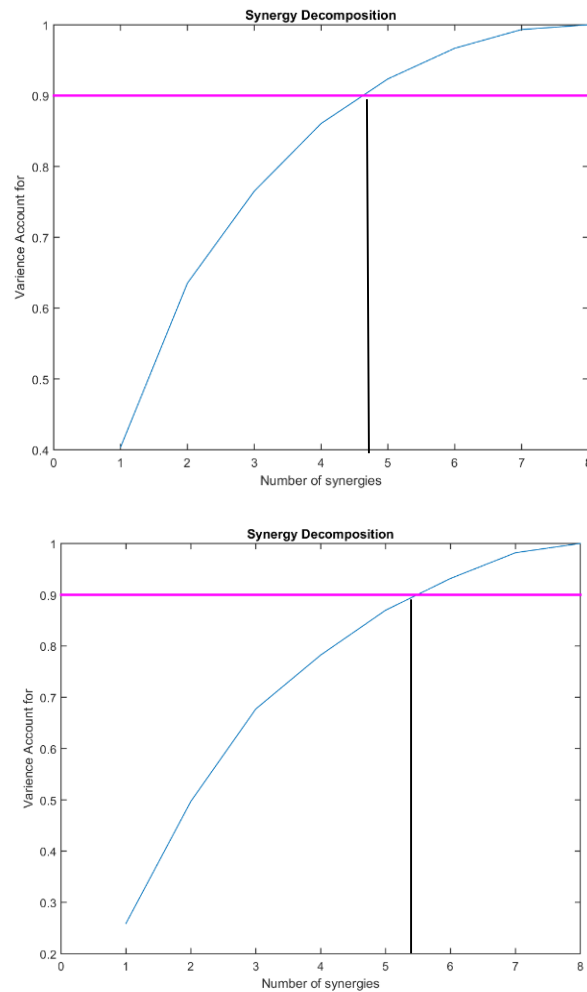
Figure 2.8 Synergy decomposition: The determination of synergy numbers. (Subj: S1) Upper: Decomposition of MEP matrix. Lower: Decomposition of ASL task related EMG matrix (Subj S1: ASL task Trial 3). Five (Left) and six (Right) synergies could explain approximately 90% variance of variability in both MEP matrix and ASL task related EMG matrix.

2.6 Synergy Matching and Comparison

A quantitative method was used to compare synergies elicited from TMS-MEP matrix and ASL-task-related EMG matrix for each subject. Dot products were computed for all pairs of synergies. For example, TMS-MEP matrix obtained from Subject S1 elicited 5 synergies. ASL-task-related EMG matrix of trial 3 of the same subject elicited 5 synergies as well. The total number of possible pairs is 5 x 5 = 25. It was done in order to find the pair that can achieve maximal correlation. Best matched pair was defined as the pair that has the highest dot product. Synergies were compared without replacement, which means once the best pair is defined, second best match should be searched from the remaining synergies that has a total number of possible pair (5-1)x(5-1) = 16. The search stopped when no possible pairs were left. The synergy comparison results are given below in Figure 2.9 on a separated page.
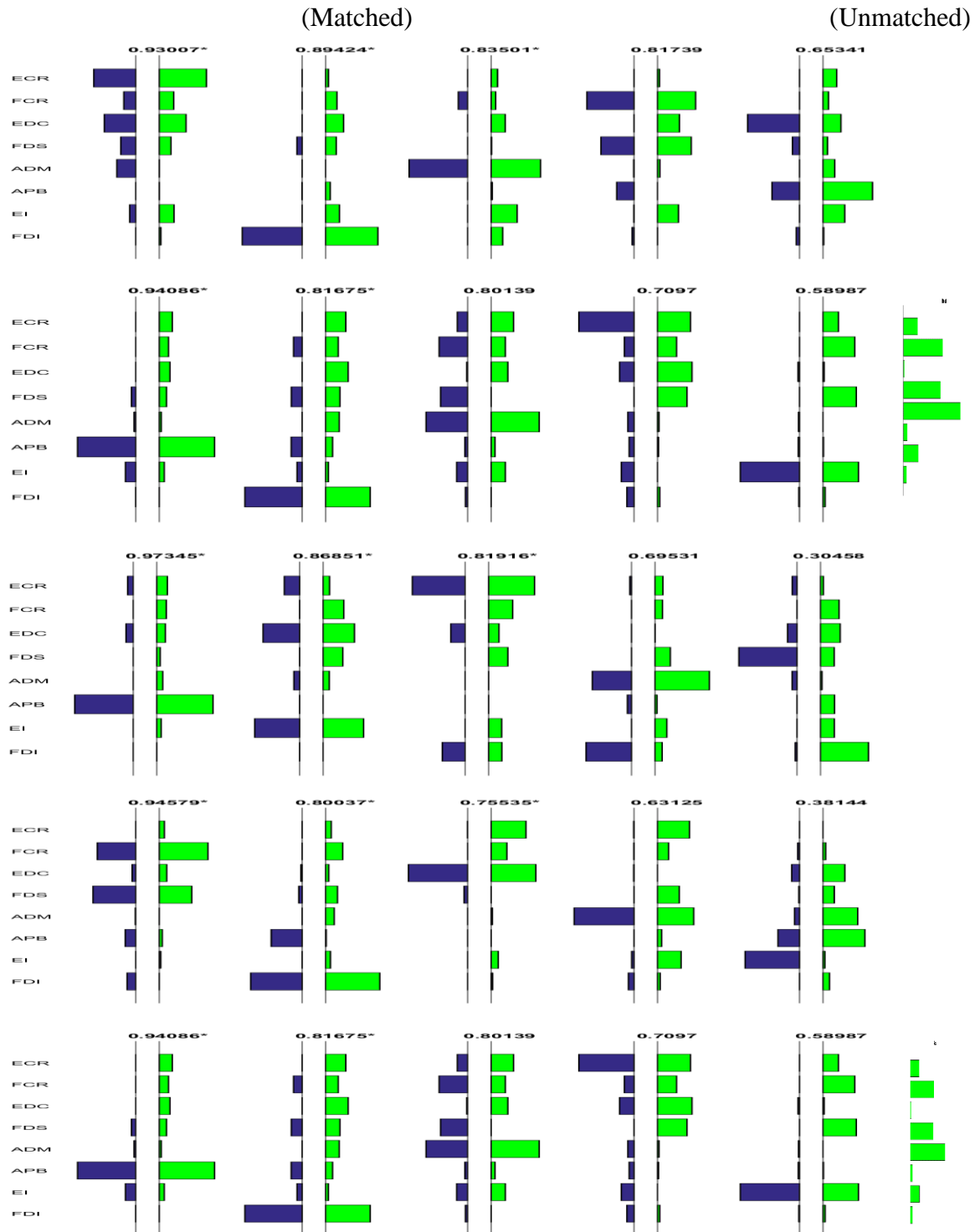
Figure 2.9 TMS-evoked muscle behavior could be decomposed into a small set of synergies mirroring those in ASL tasks. The ASL-task-related synergies (green) are shown paired together with the corresponding best-matching TMS-derived synergies (blue) ordered left to right by decreasing the dot products (numbers above the paired bar plots asterisks indicate significant correlations (p < 0.05).

2.7 Predictive Power of Synergies

The purpose of reconstruction is to see how well synergies elicited from TMS-MEP matrix or ASL task related matrix can predict the original dataset and random datasets. In this procedure, MEP vectors or ASL task related EMG vectors were reconstructed by solving the least-squares fitting problem $\min_{x} ||C * x - d||_2^2$, where x $\geq$ 0 by using a nonnegative least-squares constraint algorithm lsqnonneg that is available in MATLAB. The vector d (n by m; 1 X number of muscles) stands for normalized EMG amplitudes obtained while performing a sign language letter or number or normalized MEP amplitudes of eight channel of one stimulation site or randomly generated vector of the same size. C contains the synergy matrix obtained from either the MEP dataset or EMG dataset (n by k; number of synergies x number of muscles) by the decomposition of the NMF algorithm on the datasets. Two synergy matrices were used as basis matrices to reconstruct ASL-task-related vectors for the purpose of evaluating the predictive power. The quality of reconstruction was quantified as the PVA described in Chapter 2.5.3 between C*x, the reconstructed vector and d, the ASL task related vector.

Table 2.1 The Ability of the NMF-decomposed Synergies to Reconstruct the Original Data

Subj	H_{TMS}-> V_{TMS}	H_{vol}-> V_{vol}	H_{TMS}-> V_{rand}	H_{vol}-> V_{rand}	H_{TMS}-> V_{vol}	H_{vol}-> V_{TMS}
S1	0.96	0.90	0.62	0.56	0.73	0.82
S2	0.94	0.91	0.58	0.55	0.77	0.79
S3	0.91	0.90	0.52	0.51	0.66	0.72
S4	0.96	0.95	0.46	0.52	0.69	0.75
S5	0.95	0.92	0.49	0.50	0.63	0.86
Mean	0.94	0.92	0.53	0.53	0.70	0.79
STD	0.02	0.02	0.07	0.03	0.06	0.06

As shown in Table 2.1, H represents synergies and V represents the original data. The subset Vol stands for Voluntary movement, TMS stands for TMS and Random data. Predictive power of synergies elicited from TMS and ASL related tasks were tested and quantified. Reconstruction of original data from their elicited synergies (H_{vol} -> V_{vol}, H_{TMS} -> V_{TMS}) serve as one set of control while the predication of random matrices serves as another set of control (H_{vol} -> V_{rand}, H_{TMS} -> V_{rand}). PAVs for H_{vol} -> V_{vol} and H_{TMS} -> V_{TMS} are 0.92 ± 0.02, 0.94 ± 0.02. It verified the validity of the nonnegative least-squares constraint algorithm, and PAVs for H_{vol} -> V_{rand}, H_{TMS} -> V_{rand} are 0.53 ± 0.03 and 0.53 ± 0.07, meaning synergies are not able to predictive random data. PVAs for the cross reconstructions H_{TMS} -> V_{vol} and H_{vol} -> V_{TMS} are 0.70 ± 0.06 and 0.79 ± 0.06. The results demonstrated that synergies elicited from either TMS or voluntary movement can be predictive frameworks to explain real data.

CHAPTER 3

DISCUSSION AND CONCLUSION

The results of the study showed that TMS can be used to examine the mapping between the motor cortex and muscles involved in the control of hand muscles in healthy human subjects. Muscle responses were evoked as representations of MEP at modest TMS intensities. Unlike the results provided by the study of Gentner and Classen 2006 in which TMS could generate somatotopical gradient of finger movement, our results (Figure 2.3) do not demonstrate similar gradient topographic properties for each muscle. More rigorous studies need to be performed to verify the results. For TMS is not a direct stimulation method, the topographical results may not directly imply the relationship between stimulated region and muscles that were activated due to TMS, the angles of placing the TMS coil with respect to the scalp plain may produce different stimulation results even for the same stimulation sites. However, in this study, TMS was confirmed to be capable of generating muscle responses at a moderate intensity. Those muscle responses were not large enough to trigger mechanical constriant of hands. Therefore, we can conclude that MEPs were direct results caused by the stimulations alone plus some noise.

The hybrid NMF algorithm worked well on our data. It produced decomposition results that were visually verified to be consistent. Our results showed that for all subjects, 5 or 6 synergies can explain 90% variance of the original dataset regardless of TMS-MEP data or ASL-task-related EMG data. Figure 2.9 demonstrates a high similarity between synergies elicited from TMS-MEP data and ASL-task-related EMG data. The result is similar with Overduin et al. 2012. Results given in Table 2.1 indicate that synergies of both

class work equally well in explaining either TMS-MEP dataset or ASL-task-related dataset and can serve as predictive framwork in reconstruct real data.

Future work on the current data may involve quantitative determination similarity between same ASL trials within individual subjects to determine whether these EMG data for each trial need to be analyzed individually or not. Cortical topographic distribution of elicited synergies may be produced. In addition, statistical analysis of synergies between subjects may be performed as well.

The evaluation of synergy hypothesis is still going on after so many years of study. Even though there is considerable evidence upholding this hypothesis, some result studies provided strong results that challenge the hyposthesis (Kutch et al. 2008; Valero-Cuevas et al. 2009). In addition, the neural basis of synergies still remains unknown. All of these require more rigorous and clever studies. Stroke subjects may be good candidates in studying the neural basis of synergies. Cheung et al. 2009 observed similar synergies on affected arm and unaffected arm which may imply that muscle synergies may be generated in brain stem or spinal cord and the stroke may alter neural commands from spinal cord that results in fauly muscle behaviors. We might want to see if synergies patterns elicited from either TMS or voluntary movements of patients with brain trauma or stroke alter after rehablitation as Cheung et al. 2012 demonstrates the possibility of using systergy as a physiological markers of the status of stroke patients or patients with trauma.

My personal viewspoints regarding the field are given here. Compaired with cognative behaviors, motor behaviors are more direct therefore easier to study. One basic rationale in designing experiments is to minimize assumptions and control as many variables as possible. It is easy to apply constraints to motor behavior therefore the field of

motor control has been a hotspot for many years. However, it is inevitable to circumvent the study of the neural system when studying motor control: How neural commands are encoded and sent to spinal cord to result in movement; How these Synergy hypothsis is a good hypothesis in resolving the DOF problem, but as a theory, there must be assumptions or premises underneath. Therefore, Some questions may be worthy being explored such as whether there is difference in producing an adopt behavior vs a newly accquired behavior; how the brain plasticity relates to synergies; Why NMF is prefered.

Another paradox in the field of study is that traditional empiricism of the study methods has developed to a point when major breakthroughs is unlikely to occur unless new technologies or new image medalities are introduced. We have seen that empirical methods have pushed the field so far. However, it is like a process of learning vocabularies in a foreign language: gathering evidence and looking for correspondence between these words and words we already know in our native languages. The initial steps might be very fast. With about a hundred words, it seems like we "sort of" know that language and can travel in countries where this language is spoken. But, in order to truly know the language, it is inevitable to know the syntax and gramma.

Therefore, we need models in spite of there might be infinite number of models that can be used to explain the problem. Simple models are more powerful due to their unlikeliness of the overfitting problem, and they may be a backbone of a better fit theory or a special case of that better fit theory. For example, Newton's Law is a special case of Einstein's theory of relativity at macroscopic and low speed condition, but no one can deny its simplicity and in real life cases, Newton's Law is still preferred for it is easy for calculation. Likewise, synergies may also serve as a foundation or a building block for more complex and advanced theories that demonstrate synergetic property of motor control.

APPENDIX A

MATLAB SCRIPTS FOR DATA ANALYSIS

This sections contains MATLAB scripts for data analysis described in the context

Mass_Process.m: the mass processing function of the small automatically analyzing

toolbox I made. It is made for input the information of subjects.

```matlab
clear all
clc

warning('off','all')
warning


% Description: it is a mass processing main script
% ------------------------------------
% Major Path definination
Path_Name = 'C:\Users\popeyes\Desktop\Wei_work\Labwork\Subject Folders\';
Save_Path = 'C:\Users\popeyes\Desktop\Wei_work\Labwork\Results\';
% Add the folder and its subfolders to the search
addpath(genpath('C:\Users\popeyes\Desktop\Wei_work\Labwork\'))

% Subject Control
% It is supposed to send subject information to the subsequent function
% Subject Matrix
% Subject number by Trial number
%      Subject Initial Trial number Type
% i.e.        TM            3            1
%             GF            2            2
% P.S. Type 1 subject performed more signs than Type 2 subject
Sub_Mat = cell(5,3);
% Fill in with subject name
Sub_Mat{1,1} = 'CY';
Sub_Mat{2,1} = 'GF';
Sub_Mat{3,1} = 'NM';
Sub_Mat{4,1} = 'QQ';
Sub_Mat{5,1} = 'TM';
% Fill in with trial numbers
Sub_Mat{1,2} = 3;
Sub_Mat{2,2} = 2;
Sub_Mat{3,2} = 3;
Sub_Mat{4,2} = 3;
Sub_Mat{5,2} = 3;
% Fill in with Type
Sub_Mat{1,3} = 1;
Sub_Mat{2,3} = 2;
Sub_Mat{3,3} = 1;
```

```
Sub_Mat{4,3} = 1;
Sub_Mat{5,3} = 1;

% -----------------------------------------------------------------
% Turn Subject matrix into PATH matrix, pass the information to the
% subsequent function
% Path_Matrix = Sub_Matrix plus a column of Path
Path_Mat = cell(5,4);
Path_Mat(:,2:4) = Sub_Mat;
% Fill in Path_Mat with PATH
Path_Mat{1,1} = [Path_Name, Sub_Mat{1,1},'_Mapping\'];
Path_Mat{2,1} = [Path_Name, Sub_Mat{2,1},'_Mapping\'];
Path_Mat{3,1} = [Path_Name, Sub_Mat{3,1},'_Mapping\'];
Path_Mat{4,1} = [Path_Name, Sub_Mat{4,1},'_Mapping\'];
Path_Mat{5,1} = [Path_Name, Sub_Mat{5,1},'_Mapping\'];

% Execute Function of second layer
Main(Path_Mat(1,:),Save_Path) % CY
Main(Path_Mat(2,:),Save_Path)
Main(Path_Mat(3,:),Save_Path)
Main(Path_Mat(4,:),Save_Path)
Main(Path_Mat(5,:),Save_Path)
```

MAP_Preprocess.m: Preprocess of the MEP data and the TMS position data

```
function [N_MEPamp, NavData, GoodTrials] = MAP_Preprocess(tempData)
%    Preprocess the Data
%    Input: Loaded Raw Data
%    Output: N_MEPamp -> Normalized MEP Amplitude
%            NavData -> Position of stimulation sites

GoodTrials = find(tempData.MEPstruct.trialGood(:,1)==1); % index of good
trials
MEPamp = tempData.MEPstruct.MEPamp;
NavData = tempData.MEPstruct.NavCoords;
MEPamp = MEPamp(:,1:8);
MEPamp = MEPamp*1e3; %Convert to microvolt
MEPamp = MEPamp(GoodTrials,:); % Only retain GoodTrials
NavData = NavData(GoodTrials,:);
NavData = NavData./10; %Convert from mm to cm

% Normalize MEP by dividing its maximal value
% N_MEPamp = MEPamp/(max(max(MEPamp)));
N_MEPamp = MEPamp./repmat(max(MEPamp),length(MEPamp),1);

% Not going to use them
MAPxlim = [min(NavData(:,1)) max(NavData(:,1))]; % X limit
MAPylim = [min(NavData(:,2)) max(NavData(:,2))]; % Y limit

% Center and Normalize Positions
NavData(:,1) = (NavData(:,1) - min(NavData(:,1)))/(max(NavData(:,1))-
min(NavData(:,1)));
```

32

```matlab
NavData(:,2) = (NavData(:,2) - min(NavData(:,2)))/(max(NavData(:,2))-
min(NavData(:,2)));
N_MEPamp(isnan(N_MEPamp)) = 0;
end
```

Main.m: main function

```matlab
function Main(Path_Array, Save_Path)
%% ========================================================
% Part I Rest Mapping
% Load Data
tempData = 
load([Path_Array{1},Path_Array{2},'rest_preproc_autosave.mat']);

% Create a folder for current subject
Save_Path = [Save_Path,Path_Array{2}];
mkdir(Save_Path)
% Change directory
cd(Save_Path)

% Muscle Involved in Experiments
muscle = {'FDI';'EI';'APB';'ADM';'FDS';'EDC';'FCR';'ECR'};
%% --------------------------------------------------------
% Data Preprocessing
[N_MEPamp, NavData, GoodTrials] = MAP_Preprocess(tempData);
%% --------------------------------------------------------
% Data Visualization
% MAP_Visualization(N_MEPamp, NavData, GoodTrials, tempData)
%% --------------------------------------------------------
% Muscle Synergies Identification and Plot
[H_Rest, synergy_Rest] = synergy_extraction(N_MEPamp, muscle);
%% --------------------------------------------------------
% PCA Analysis and plots (need 2012b or later to run this function)
% PCA_Analy(N_MEPamp, muscle)


%% ========================================================
% Part II Voluntary movement
for Trial_Num = 1:cell2mat(Path_Array(3))
    % Make a new directory to store
    Trial_Path = [Save_Path, '\' , num2str(Trial_Num), '\'];
    mkdir(Trial_Path)
    % Change directory
    cd(Trial_Path)
    % --------------------------------------------------
    %
    Vol_Data = Vol_Mov_Preprocess(Path_Array, Trial_Num);
    % Calculate
    [H_Mov, synergy_Mov] = synergy_extraction(Vol_Data.N_MEPamp,
Vol_Data.Muscle);

    %% ========================================================
```

```matlab
    % Part III Find Best Match
    [match_mat,max_Dot_Prod_All,p_value,match_num] =
Best_Match(muscle,synergy_Rest, synergy_Mov, H_Rest, H_Mov);

    %
------------------------------------------------------------------------
-------
    % Change directory back to upper level
    cd(Save_Path)

    % Publish this result to excel spreadsheet
    filename = 'results.xlsx';
    sheet = Trial_Num;
    xlswrite(filename, {'Index of matched column'}, sheet, 'B1');
    xlswrite(filename, {'H_Rest'}, sheet, 'E1');
    xlswrite(filename, {'H_Mov'}, sheet, 'F1');
    xlswrite(filename, {'Dot Product'}, sheet, 'G1');
    xlswrite(filename, {'Pearson r correlation'}, sheet, 'H1');
    xlswrite(filename, match_mat ,sheet, ['E2:F',num2str(match_num+1)]);
    xlswrite(filename, max_Dot_Prod_All, sheet,
['G2:G',num2str(match_num+1)])
    xlswrite(filename, p_value, sheet, ['H2:H',num2str(match_num+1)])



    % The Mante Carlo simulation isn't quite important for now
    % % Find the significance
    % % Run simulation
    % MAX_DOT = Significance(N_MEPamp, Vol_Data.N_MEPamp);
    % X = 1:5;
    % for m = 1:5
    %     X(m) = prctile(MAX_DOT(m,:),95);% 95 percentile
    % end
    % % We can maunually to see if they are significant
    % $$$$$$$

    %% ==========================================================
    % Part IV Reconstruction and Error
    % Synergies elicited from MEP reconstruct MEP (Rest->Rest)
    [Error_Vec_Rest_Rest, PVE_Vec_Rest_Rest, VAF_Vec_Rest_Rest] =
Assessment_Calc(H_Rest,N_MEPamp,synergy_Rest);
    % Synergies elicited from ASL tasks reconstruct ASL tasks (ASL->ASL)
    [Error_Vec_Vol_Vol, PVE_Vec_Vol_Vol, VAF_Vec_Vol_Vol] =
Assessment_Calc(H_Mov,Vol_Data.N_MEPamp,synergy_Mov);
    % Synergies elicited from MEP reconstruct ASL tasks (Rest->ASL)
    [Error_Vec_Rest_Vol, PVE_Vec_Rest_Vol, VAF_Vec_Rest_Vol] =
Assessment_Calc(H_Rest,Vol_Data.N_MEPamp,synergy_Mov);
    % Synergies elicited from ASL tasks reconstruct MEP (ASL->Rest)
    [Error_Vec_Vol_Rest, PVE_Vec_Vol_Rest, VAF_Vec_Vol_Rest] =
Assessment_Calc(H_Mov,N_MEPamp,synergy_Rest);
    % ----------------------------------------------------
    % Random Matrix with the same size as MEP
    Rand_Mat = rand(size(N_MEPamp));
    % Synergies elicited from MEP reconstruct Random Matrix (Rest->Random)
    [Error_Vec_Rest_Ran, PVE_Vec_Rest_Ran, VAF_Vec_Rest_Ran] =
Assessment_Calc(H_Rest,Rand_Mat,synergy_Rest);
```

```matlab
     % Random Matrix with the same size as MEP
    Rand_Mat = rand(size(Vol_Data.N_MEPamp));
    % Synergies elicited from ASL tasks reconstruct Random Matrix
(ASL->Random)
    [Error_Vec_Vol_Ran, PVE_Vec_Vol_Ran, VAF_Vec_Vol_Ran] =
Assessment_Calc(H_Mov,Rand_Mat,synergy_Mov);




    % Calculate mean error and std of RMSE
    % A 18 by 2 matrix
    %              Mean    Standard Deviation
    % Rest->Rest
    % ASL->ASL
    % Rest->ASL
    % ASL->Rest
    % Rest->Ran
    % ASL->Ran

    Results = [mean(Error_Vec_Rest_Rest) std(Error_Vec_Rest_Rest);...
        mean(Error_Vec_Vol_Vol) std(Error_Vec_Vol_Vol);...
        mean(Error_Vec_Rest_Vol) std(Error_Vec_Rest_Vol);...
        mean(Error_Vec_Vol_Rest) std(Error_Vec_Vol_Rest);...
        mean(Error_Vec_Rest_Ran) std(Error_Vec_Rest_Ran);...
        mean(Error_Vec_Vol_Ran) std(Error_Vec_Vol_Ran);...%%%%
        mean(PVE_Vec_Rest_Rest) std(PVE_Vec_Rest_Rest);...
        mean(PVE_Vec_Vol_Vol) std(PVE_Vec_Vol_Vol);...
        mean(PVE_Vec_Rest_Vol) std(PVE_Vec_Rest_Vol);...
        mean(PVE_Vec_Vol_Rest) std(PVE_Vec_Vol_Rest);...
        mean(PVE_Vec_Rest_Ran) std(PVE_Vec_Rest_Ran);...
        mean(PVE_Vec_Vol_Ran) std(PVE_Vec_Vol_Ran);...%%%%
        mean(VAF_Vec_Rest_Rest) std(VAF_Vec_Rest_Rest);...
        mean(VAF_Vec_Vol_Vol) std(VAF_Vec_Vol_Vol);...
        mean(VAF_Vec_Rest_Vol) std(VAF_Vec_Rest_Vol);...
        mean(VAF_Vec_Vol_Rest) std(VAF_Vec_Vol_Rest);...
        mean(VAF_Vec_Rest_Ran) std(VAF_Vec_Rest_Ran);...
        mean(VAF_Vec_Vol_Ran) std(VAF_Vec_Vol_Ran)]; %%%%

    % Publish this result to excel spreadsheet
    xlswrite(filename, {'First 6 rows: RMSE, Second 6 rows: PVE,  Third 6
rows: VAF'}, sheet, 'E10');
    xlswrite(filename,
{'Rest->Rest';'ASL->ASL';'Rest->ASL';'ASL->Rest';'Rest->Random';'ASL->R
andom'}, sheet, 'E11:E16');
    xlswrite(filename,
{'Rest->Rest';'ASL->ASL';'Rest->ASL';'ASL->Rest';'Rest->Random';'ASL->R
andom'}, sheet, 'E17:E22');
    xlswrite(filename,
{'Rest->Rest';'ASL->ASL';'Rest->ASL';'ASL->Rest';'Rest->Random';'ASL->R
andom'}, sheet, 'E23:E28');
    xlswrite(filename, Results,sheet,'F11:G28');


end
```

MAP_Visualization.m: Visualization of TMS positions and their corresponding MEPs of various muscles.

```matlab
function MAP_Visualization(N_MEPamp, NavData, GoodTrials, tempData)
%   Visualize
%   Input: Loaded Raw Dat
muscle = {'FDI';'EI';'APB';'ADM';'FDS';'EDC';'BIC';'FDI OPP'};


% Plot all trials (Normalized MEP)
h_fig = figure('Visible', 'off');
subplot(2,2,1)
plot(NavData(:,1),NavData(:,2),'k','Marker','o','LineStyle','none')
title('Normalized Stimulation Sites (x-y plane)')
xlabel('x position')
ylabel('y position')

% Plot all trials (Normalized MEP)
subplot(2,2,2)
N_MEPamp(isnan(N_MEPamp)) = 0;
plot(N_MEPamp')
xlim([0 9])
set(gca,'XTick',1:8)
set(gca,'XTickLabel',muscle);
title('MEP Tuning Curves')
xlabel('Muscles')
ylabel('Normalized MEPs')

% Plot randomly selected stimulation sites
n = randi([1 length(GoodTrials)],1,3); % 3 random integer
subplot(2,2,3)
plot(NavData(n,1),NavData(n,2),'k','Marker','o','LineStyle','none')%
xlim([0, 1])
ylim([0, 1])
title('Normalized Randomly Selected Stimulation Sites (x-y plane)')
xlabel('x position')
ylabel('y position')

% Plot randomly selected trials (Normalized MEP)
subplot(2,2,4)
plot(N_MEPamp(n,:)')
xlim([0 9])
set(gca,'XTick',1:8)
set(gca,'XTickLabel',muscle);
title('Corresponding MEP Tuning Curves')
xlabel('Muscles')
ylabel('Normalized MEPs')

saveas(h_fig, 'Visualization_1.png')
close(h_fig)
```

```matlab
%% ------------------------------------------------------------------------
% Plot EMG for selected Trials
h_fig = figure('Visible', 'off');
for i = 1:8
    for j = 1:3
        subplot(8,3,j+(i-1)*3)
        plot(tempData.MEPstruct.EMG{n(j),i})
        xlabel('Time')
        ylabel(muscle(i))
        %ylim([-0.1, 0.1])
    end
end
saveas(h_fig, 'Visualization_2.png')
close(h_fig)


%-------------------------------------------------------------------------
% Plot stem figure
h_fig = figure('Visible', 'off');
for k = 1:8
    subplot(2,4,k)
    stem3(NavData(:,1),NavData(:,2),N_MEPamp(:,k))
    title(muscle(k))
end
% Intrapolant figures
xlin = linspace(min(NavData(:,1)),max(NavData(:,1)),100);
ylin = linspace(min(NavData(:,2)),max(NavData(:,2)),100);
[X, Y] = meshgrid(xlin,ylin);
saveas(h_fig, 'Visualization_3.png')
close(h_fig)

h_fig = figure('Visible', 'off');
for l = 1:8
    subplot(2,4,l)
    f = scatteredInterpolant(NavData(:,1),NavData(:,2),N_MEPamp(:,l));
    Z = f(X,Y);
    mesh(X,Y,Z)
    zlim([0,1])
    axis tight; hold on
    plot3(NavData(:,1),NavData(:,2),N_MEPamp(:,l),'.','MarkerSize',15)
%nonuniform
    title(muscle(k))
end
saveas(h_fig, 'Visualization_4.png')
close(h_fig)

% Covariance matrix
Cov_Mat = cov(N_MEPamp);
h_fig = figure('Visible','off');
imagesc(Cov_Mat)
axis ij
axis square
title('Covarience Matrix')
colorbar
set(gca,'XTickLabel',muscle)
set(gca, 'YTickLabel',muscle)
```

```
saveas(h_fig, 'Covariance.png')
close(h_fig)

% Correlation matrix
Cor_Mat = corrcov(Cov_Mat);
h_fig = figure('Visible','off');
imagesc(Cor_Mat)
axis ij
axis square
title('Correlation Matrix')
colorbar
set(gca,'XTickLabel',muscle)
set(gca, 'YTickLabel',muscle)

saveas(h_fig, 'Correlation.png')
close(h_fig)

end
```

Vol_Mov_Preprocess.m: Preprocess of the ASL task EMG data.

```
function Preprocessed_Data = Vol_Mov_Preprocess(Path_Array, Trial_Num)
% Preprocess voluntary movement data
%   Load EMG Data and Preprocess to generate and return amplitude data
%   Each subject perform sign language number and letter

% Sign Name
switch Path_Array{4}
    case 1
        % For subject TM, NM, QQ, CY 32 signs
        Sign_Name = {'A';'B';'C';'E';'F';'G';'H';'I';'K';'L';'M';'N';...
            'O';'P';'Q';'R';'S';'T';'U';'V';'W';'X';'Y';...
            '1';'2';'3';'4';'5';'6';'7';'8';'9'};
    case 2
        % For subject GF 28 signs
        Sign_Name = {'A';'B';'C';'E';'F';'G';'H';'I';'K';'L';'M';'N';...
            'O';'P';'Q';'R';'S';'T';'U';'V';'W';'X';'Y'...
            ;'1';'2';'3';'4';'5'};
end
Num_Sign = length(Sign_Name);
% Muscle Name
muscle = {'FDI';'EI';'APB';'ADM';'FDS';'EDC';'FCR';'ECR'};
Num_Muscle = length(muscle);
% Path

Path_Name = [Path_Array{1}, '\ASLtask\Trial', num2str(Trial_Num), '\'];

%
-----------------------------------------------------------------------
-----------
Preprocessed_Data = []; % structure to load EMG data

% EMG filter profile
```

```matlab
fs = 1000; % sampling frequency
fc_H = 10; % cutoff frequency High Pass
fc_L = 300; %cutoff frequency Low Pass
order = 6; % filter order
nfc_H = 2*fc_H/fs; % normalized cutoff frequency
nfc_L = 2*fc_L/fs; % normalized cutoff frequency
[B_H,A_H] = butter(order, nfc_H,'high');
[B_L,A_L] = butter(order, nfc_L,'low');
[B_Power,A_Power] = butter(2, [59.5 60.5]*2/fs,'stop'); % Power remove
%
-------------------------------------------------------------------------
------
h_fig = figure('Visible', 'off');
plotscale = 0.8; % to make all EMG on the same subplot
MEP_amp = zeros(Num_Sign, Num_Muscle);% Pre-defined matrix to store
Amplitude
for i = 1:Num_Sign
    % load files to create a structure
    temp = load([Path_Name,Sign_Name{i},'.mat']); % load data
    temp_EMG = temp.EMG;
    time = (1:length(temp_EMG(:,1)))/fs; % length of EMG/fs = time

    subplot(2, ceil(Num_Sign/2), i)
    % Filter each muscle
    for j = 1:Num_Muscle
        temp_EMG(:,j) = filtfilt(B_H, A_H, temp_EMG(:,j));
        temp_EMG(:,j) = filtfilt(B_L, A_L, temp_EMG(:,j));
        temp_EMG(:,j) = filtfilt(B_Power, A_Power, temp_EMG(:,j));
        temp_EMG(:,j) = abs(temp_EMG(:,j));
        % Calculate Amplitude and Normalized Amplitude
        a = temp_EMG(:,j);
        MEP_amp(i,j) = mean(a(5000:7000));
        plot(time, temp_EMG(:,j) + j*plotscale)
        hold on
    end
    % I need to put ylabel and xlabel here to make the plot look better
    set(gca, 'XTickLabel',[])
    set(gca, 'YTickLabel',[])

end
saveas(h_fig, 'ASL_EMG.png')
close(h_fig)


% Normalize MEP_amp
N_MEPamp = MEP_amp./repmat(max(MEP_amp),length(MEP_amp),1);
%%%% Note all sign amp stay here

%
-------------------------------------------------------------------------
% Draw circles
h_fig = figure('Visible', 'off');
for i = 1:Num_Sign
    subplot(4, ceil(Num_Sign/4), i)
    circleplot(N_MEPamp(i,:), muscle, Sign_Name(i))
end
saveas(h_fig, 'ASL_AMP.png')
```

39

```matlab
close(h_fig)


% Store data in a structure for output
Preprocessed_Data.Sign = Sign_Name;
Preprocessed_Data.Muscle = muscle;
Preprocessed_Data.MEP_amp = MEP_amp;
Preprocessed_Data.N_MEPamp = N_MEPamp;
end
```

synergy_extraction.m: Extract synergy matrix

```matlab
function [H, synergy_num] = synergy_extraction( N_MEPamp, muscle )
%Extraction of muscle synergies using NNMF algorithm
% Muscle synergy (NMF)
% Identify number of synergies
% Using MATLAB built-in function nnmf
% ----------------------------------------------------------
% Determine if one array in the matrix are equal to zero
% If so, delete them (Decrease the sparseness
N_MEPamp(find(any(N_MEPamp')==0),:) = [];
[muscle_num, ~] = size(muscle);

R2 = 1:muscle_num;
VAF = 1:muscle_num;

for i = 1:muscle_num
    % A possible solution to find more consistent solutions
    opt = statset('MaxIter',100,'Display','final');
    % Maximal iteration allowed = 100; Display final outcome
    [W0,H0] =
nnmf(N_MEPamp,i,'replicates',10,'options',opt,'algorithm','mult');
    opt = statset('Maxiter',10000,'Display','final');
    [W,H,D] =
nnmf(N_MEPamp,i,'w0',W0,'h0',H0,'options',opt,'algorithm','als');
    % Decompose normalized MEP Amplitude
    % N_MEPamp: 80by8 => W:80*i & H:i*8
    R2(i) = D;
    % ---------------------------------------------
    % Calculate VAF by using the formula on the webpage
    % http://www.rasch.org/rmt/rmt173g.htm
    % VAF = 1 - UV/OV
    % UV = (x-y)'*(x-y) --- we only want numbers on the diagonal
    % OV = (y-mean(y))'*(y-mean(y)) --- The same as above, we only want
    % numbers on the diagonal
    % ---------------------------------------------
    temp = N_MEPamp - W*H;
    UV = trace(temp*temp');
    temp_2 = bsxfun(@minus,N_MEPamp,mean(N_MEPamp, 2));
    OV = trace(temp_2*temp_2');
    VAF(i) = 1 - UV/OV;
end
```

```matlab
% Set the threshold to 0.05
% 1.Residual method
% [~, synergy_num] = min(abs(R2-0.05)); % find the nearest
% 2.VAF method
[~, synergy_num] = min(abs(VAF-0.9)); % find the nearest set to 90%


%% ----------------------------------------------------------------
h_fig = figure('Visible', 'off');
plot(R2)
xlabel('Number of synergies')
ylabel('Root-mean-squared residual')
title('Synergy Decomposition')
line('XData', [0 muscle_num], 'YData', [0.05 0.05], 'LineStyle', '-', ...
    'LineWidth', 2, 'Color','m');
saveas(h_fig, 'Synergy_Decomp.png')
close(h_fig)
% ----------------------------------------------------------------
h_fig = figure('Visible', 'off');
plot(VAF)
xlabel('Number of synergies')
ylabel('Varience Account for')
title('Synergy Decomposition')
line('XData', [0 muscle_num], 'YData', [0.9 0.9], 'LineStyle', '-', ...
    'LineWidth', 2, 'Color','m');
saveas(h_fig, 'Synergy_Decomp_2.png')
close(h_fig)


%% --------------------------------------------------------------------
% A possible solution to find more consistent solutions
opt = statset('MaxIter',100,'Display','final');
[W0,H0] =
nnmf(N_MEPamp,synergy_num,'replicates',10,'options',opt,'algorithm','mu
lt');
% Maximal iteration allowed = 100; Display final outcome
% The number of times to repeat the factorization, using new random
% starting values for W and H. "Mult" represents the multiplicative update
% algorithm that is more sensitive to initial values. "Replicates" can be
% used to find W and H from multiple random starting values.
% The command above gives out 10 iteration, and the best one (smallest RMS)
% is used to feed for alternating least squares for more iteration.

opt = statset('Maxiter',10000,'Display','final');
[~,H,~] =
nnmf(N_MEPamp,synergy_num,'w0',W0,'h0',H0,'options',opt,'algorithm','al
s');
% Maximal interation allowed = 10000; Display final outcome
% "als" stands for the alternating least squares algorithm that converges
% faster and more consistently.
% Final # of synergies is synergy_num
% H contains some synergies; W contains weights for Good trials


% -------------------------------------------------------------------
% Visualize synergies by muscles (We are interested in H)
h_fig = figure('Visible', 'off');
bar(H');
set(gca,'XTickLabel',muscle);
```

```matlab
xlabel('Muscles')
ylabel('Normalized MEP')
title('Synergies Visualization')
ylim([0 2])
legend_title = cell(synergy_num,1);
for j = 1:synergy_num
    legend_title(j) = cellstr(['Synergy ',num2str(j)]);
end
legend(legend_title)
saveas(h_fig, 'synergy_by_muscle.png')
close(h_fig)



% Visualize synergies by synergies
h_fig = figure('Visible', 'off');
bar(H);
set(gca,'XTickLabel',legend_title);
xlabel('Synergies')
ylabel('Normalized MEP')
title('Synergies Visualization')
ylim([0 2])
legend(muscle)
saveas(h_fig, 'synergy_by_synergy.png')
close(h_fig)



end
```

synergy_extraction_no_plot.m: extract synergies matrix but not

```matlab
function [H, synergy_num] = synergy_extraction_no_plots(N_MEPamp, muscle)
%Extraction of muscle synergies using NNMF algorithm
% Muscle synergy (NMF)
% Identify number of synergies
% Using MATLAB built-in function nnmf
% Not produce plots only for calculation

% ---------------------------------------
% Determine if one array in the matrix are equal to zero
% If so, delete them (Decrease the sparseness
N_MEPamp(find(any(N_MEPamp')==0),:) = [];
[muscle_num, ~] = size(muscle);

R2 = 1:muscle_num;
VAF = 1:muscle_num;

for i = 1:muscle_num
    % A possible solution to find more consistent solutions
    opt = statset('MaxIter',100,'Display','final');
    % Maximal iteration allowed = 100; Display final outcome
    [W0,H0] =
nnmf(N_MEPamp,i,'replicates',10,'options',opt,'algorithm','mult');
    opt = statset('Maxiter',10000,'Display','final');
```

```matlab
    [W,H,D] =
nnmf(N_MEPamp,i,'w0',W0,'h0',H0,'options',opt,'algorithm','als');
    % Decompose normalized MEP Amplitude
    % N_MEPamp: 80by8 => W:80*i & H:i*8
    R2(i) = D;
    % -----------------------------------------------
    % Calculate VAF by using the formula on the webpage
    % http://www.rasch.org/rmt/rmt173g.htm
    % VAF = 1 - UV/OV
    % UV = (x-y)'*(x-y) --- we only want numbers on the diagonal
    % OV = (y-mean(y))'*(y-mean(y)) --- The same as above, we only want
    % numbers on the diagonal
    % -----------------------------------------------
    temp = N_MEPamp - W*H;
    UV = trace(temp*temp');
    temp_2 = bsxfun(@minus,N_MEPamp,mean(N_MEPamp, 2));
    OV = trace(temp_2*temp_2');
    VAF(i) = 1 - UV/OV;
end


% Set the threshold to 0.05
% 1.Residual method
% [~, synergy_num] = min(abs(R2-0.05)); % find the nearest
% 2.VAF method
[~, synergy_num] = min(abs(VAF-0.9)); % find the nearest set to 90%



% A possible solution to find more consistent solutions
opt = statset('MaxIter',100,'Display','final');
[W0,H0] =
nnmf(N_MEPamp,synergy_num,'replicates',10,'options',opt,'algorithm','mu
lt');
% Maximal iteration allowed = 100; Display final outcome
% The number of times to repeat the factorization, using new random
% starting values for W and H. "Mult" represents the multiplicative update
% algorithm that is more sensitive to initial values. "Replicates" can be
% used to find W and H from multiple random starting values.
% The command above gives out 10 iteration, and the best one (smallest RMS)
% is used to feed for alternating least squares for more iteration.

opt = statset('Maxiter',10000,'Display','final');
[~,H,~] =
nnmf(N_MEPamp,synergy_num,'w0',W0,'h0',H0,'options',opt,'algorithm','al
s');
% Maximal interation allowed = 10000; Display final outcome
% "als" stands for the alternating least squares algorithm that converges
% faster and more consistently.




% Final # of synergies is 5
% H contains 5 synergies; W contains weights for Good trials
end
```

Best_Match.m: Find the best match

```matlab
function [match_mat,max_Dot_Prod_All,p_value,match_num] =
Best_Match(muscle, synergy_Rest, synergy_Mov, H_Rest, H_Mov)
%UNTITLED3 Summary of this function goes here
%   Detailed explanation goes here

[muscle_size, ~] = size(muscle);
% Compute Dot Product
    Dot_prod = zeros(synergy_Rest,synergy_Mov); % syngery_Rest x
synergy_Mov x axis: Rest; y axis: Vol
    for i = 1:synergy_Rest
        for j = 1:synergy_Mov
        Dot_prod(i,j) = sum(H_Rest(i,:).*H_Mov(j,:));
        end
    end

    % Find largest dotproduct to determine the best match
    % Determine the significance
    % Calculated using Pearson's r correlation
    % http://faculty.quinnipiac.edu/libarts/polsci/Statistics.html
    %
https://en.wikipedia.org/wiki/Pearson_product-moment_correlation_coeffi
cient
    match_num = min(synergy_Rest, synergy_Mov);
    match_mat = zeros(match_num,2); % Store match index column 1:rest;
col2:mov
    max_Dot_Prod_All = zeros(match_num,1); % Store Dot product
    p_value = zeros(match_num,1); % Store P value
    % The following for loop does three jobs
    % 1: finding matches 2: calculating p values 3: plot matching
    h_fig = figure('Visible', 'off');
    for k = 1:match_num
        max_dot = max(max(Dot_prod));
        max_Dot_Prod_All(k,1) = max_dot;
        [r,c] = find(Dot_prod == max_dot);
        match_mat(k,1) = r; % match
        match_mat(k,2) = c; % match
        [~,x] = corrcoef(H_Rest(r,:), H_Mov(c,:)); % calculate p value
        % t test here
        % [h,~,~,~] = ttest2(H_Rest(r,:), H_Mov(c,:));
        p_value(k) = x(2,1);
        % clear corresponding row and column
        Dot_prod(r,:) = 0;
        Dot_prod(:,c) = 0;

        % Plot
        subplot(1,2*match_num,(k-1)*2+1)
        h = barh(H_Rest(r,:)');
        set(get(h,'Parent'),'xdir','r')
        axis off
        if p_value(k) <= 0.05
```

```matlab
            h_1 = title([num2str(max_dot),'*']);
        else
            h_1 = title(num2str(max_dot));
        end
        P = get(h_1,'Position');
        set(h_1,'Position',[P(1)-1.05 P(2) P(3)])
         if k == 1
            for n = 1:muscle_size
                text(2.1,n,muscle(n))
            end
         end
         subplot(1,2*match_num,2*k)
         barh(H_Mov(c,:)')
         xlim([0 1])
         axis off
    end
    saveas(h_fig, 'matching.png')


    % Plot unmatched synergies
    % Three cases: synergy_Rest > synergy_Mov
    %              synergy_Rest < synergy_Mov
    %              synergy_Rest == synergy_Mov
    if synergy_Rest < synergy_Mov
        h_fig_1 = figure('Visible', 'off');
        unmatched = find(ismember((1:synergy_Mov),match_mat(:,2))==0);
        unmatched_num = synergy_Mov - synergy_Rest;
        for o = 1:unmatched_num
            subplot(1,unmatched_num,o)
            barh(H_Mov(unmatched(o),:)')
            if o == 1
            for p = 1:muscle_size
                text(1,p,muscle(p))
            end
            end
            title('Mov')
            xlim([0 1])
            axis off
        end
        saveas(h_fig_1, 'unmatching.png')
    elseif synergy_Rest > synergy_Mov
        h_fig_1 = figure('Visible', 'off');
        unmatched = find(ismember((1:synergy_Rest),match_mat(:,2))==0);
        unmatched_num = synergy_Rest - synergy_Mov;
        for o = 1:unmatched_num
            subplot(1,unmatched_num,o)
            barh(H_Rest(unmatched(o),:)')
            if o == 1
            for p = 1:muscle_size
                text(-0.5,p,muscle(p))
            end
            end
            title('Rest')
            xlim([0 1])
            axis off
        end
        saveas(h_fig_1, 'unmatching.png')
    end
```

```
end
```

Assessment_Calc.m: Calculate PVA

```
function [Error, PVE, VAF] = Assessment_Calc(H,N_MEPamp,synergy_num)
% Input:
% H: synergies elicited
% N_MEPamp: Matrix ready for reconstruction
% synergy_num: # of synergies
% //////////////////////////////
% Output:
% Error: Root-mean-squared residual
% PVE: Proportion of variance explained
% VAF: Varience account for
% Equations are given below
% //////////////////////////////


%% --------------------------------------------------------
Num = length(N_MEPamp); % Number of rows
Error = 1: Num; % Error vector
% -----------------------------------------------------------------
PVE = 1: Num; % PVE vector
% -----------------------------------------------------------------
VAF = 1: Num; % VAF vector
% -----------------------------------------------------------------
%%
for m = 1:Num
    % Version I Root Mean Squared Error (RMSE)
    % lsqnonneg:Solve nonnegative least-squares constraint problem
    % Error = norm(d-C*X)/sqrt(n*m) same as determined D calculated in NMF
function
    % Here n = 1; m = synergy_num
    % -----------------------------------------------------------
    % Version II Proportion of Variance Explained (PVE)
    % The explanation of PVE is given at:
    % https://www.msu.edu/user/sw/statrev/strv211.htm
    %
https://en.wikipedia.org/wiki/Explained_variation#Correlation_coefficie
nt_as_measure_of_explained_variance
    % PVE = SSR/SST
    % SSR: (x-mean(x))'*(y-mean(y)))^2
    % SST: (x-mean(x))'*(x-mean(x))*((y-mean(y))'*(y-mean(y)))
    % ------------------------------------------------
    % Version III Variance Account for (VAF)
    % http://www.rasch.org/rmt/rmt173g.htm (Website was provided by Mat)
    % VAF = 1 - UV/OV
    % UV = (x-y)'*(x-y)
    % OV = (y-mean(y))'*(y-mean(y))
    % -----------------------------------------------------------
```

```matlab
    % Calculate X: non-negative least squared fit (nnlsf) (save computing
power)
    % Calculate C*X Reconstructed muscle response for a single trial
    CX = transpose(H)*lsqnonneg(transpose(H),transpose(N_MEPamp(m,:)));

    % -------------------------------------------------------
    % Voluntary activity -> Voluntary activity
    Error(m) = norm(CX - transpose(N_MEPamp(m,:)))/sqrt(synergy_num);
    %%%%%%%%%%%%%%%%%
    PVE(m) = ((CX - mean(CX))'...
        *(transpose(N_MEPamp(m,:))-mean(transpose(N_MEPamp(m,:))))^2
...
        /((CX-mean(CX))'*(CX-mean(CX))...

*((transpose(N_MEPamp(m,:)))-mean(transpose(N_MEPamp(m,:))))'*(transpos
e(N_MEPamp(m,:))-mean(transpose(N_MEPamp(m,:)))));
    %%%%%%%%%%%%%%%%%
        VAF(m) = 1 - (CX -transpose(N_MEPamp(m,:)))'*(CX
-transpose(N_MEPamp(m,:)))...

/(((transpose(N_MEPamp(m,:)))-mean(transpose(N_MEPamp(m,:))))'*(transpo
se(N_MEPamp(m,:))-mean(transpose(N_MEPamp(m,:)))));

end

PVE(isnan(PVE)) = [];
Error(find(Error)==0) = [];
VAF(isnan(VAF)) = [];

end
```

PCA_Analy.m: PCA Analysis

```matlab
function PCA_Analy(N_MEPamp, muscle)
%PCA Analysis


% Distance of MP's MV
n = length(N_MEPamp);
Distance = zeros(n, n);
MyDistance = zeros(n, n);
for i = 1:n
    temp = sqrt(sum(N_MEPamp(i,:).*N_MEPamp(i,:)));
    temp0 = N_MEPamp(i,:);
    for j = 1:n
        % d_MV = 1 - sum(MVi*MVj)/sqrt
        Distance(i,j) = 1 - sum(N_MEPamp(i,:).*N_MEPamp(j,:))...
            /(temp*sqrt(sum(N_MEPamp(j,:).*N_MEPamp(j,:))));
        % My distance is just view them as a point in high dimensional
        % space, and find their natural distance
        temp1 = temp0 - N_MEPamp(j,:);
```

```matlab
            MyDistance(i,j) = sqrt(sum(temp1.*temp1));
        end
end
% Normalize MyDistance
MyDistance = MyDistance./(max(max(MyDistance)));

% Distance matrices visualization
h_fig = figure('Visible', 'off');
pcolor(MyDistance)
axis ij
axis square
title('Distance')
colorbar
saveas(h_fig, 'PCA_Dist_Mat.png')
close(h_fig)


%% -------------------------------------------------------------
%%
% =====================================================================
% Hierachical cluster analysis is used to cluster muscle vector (MV)
% Distance calculated here is euclidean distance
% figure
% Z_1 = linkage(pdist(N_MEPamp));
% dendrogram(Z_1)
% title('Hierachical clustering')

% PCA
[PCAcoeff, PCAscore, eigenvalues,~, explained] = pca(N_MEPamp,...
    'VariableWeights','Variance');

% Scree Plot
% figure
% plot(eigenvalues);
% title('Scree Plot')
% line('XData', [0 8], 'YData', [1 1], 'LineStyle', '-', ...
%     'LineWidth', 2, 'Color','m');

h_fig = figure('Visible', 'off');
pareto(explained)
xlabel('Principal Component')
ylabel('Variance Explained (%)')
title('Variance Explained')
saveas(h_fig, 'PCA_Explained.png')
close(h_fig)


% Plot of component weights
% figure
% marker = '.ox+*sdv';
% for p = 1:8
%     line('XData', [0 PCAcoeff(p,1)], 'YData', [0
PCAcoeff(p,2)],'Marker',marker(p),'LineStyle', '-', ...
%     'LineWidth', 2, 'Color', 'b');
%     legend(muscle(p))
%     hold on
```

```matlab
% end

% A better plot - Biplot
[numobs, numvars] = size(N_MEPamp);
h_fig = figure('Visible', 'off');
hold on;
scatter(PCAscore(:,1), PCAscore(:,2), 5, 'bo');
scale = max((abs(PCAscore(:,1:2)))) ./ max((abs(PCAcoeff(:,1:2))));
scatter(PCAcoeff(:,1)*scale(1), PCAcoeff(:,2)*scale(2), 'ro', 'filled');
for i = 1:numvars
    line([0,PCAcoeff(i,1)*scale(1)],...
        [0,PCAcoeff(i,2)*scale(2)], 'Color', 'r');
    text(PCAcoeff(i,1)*scale(1)-1, PCAcoeff(i,2)*scale(2), muscle{i});
end
title('Principal Component Biplot');
xlabel('PC 1');
ylabel('PC 2');
saveas(h_fig, 'PCA_Biplot.png')
close(h_fig)


end
```

circleplot.m: Plot MV circles

```matlab
function circleplot(N_MEPamp, muscle, Sign_Name)
%Produce circleplot for each sign movement
%    Inputs are muscles and N_MEPamp
mean_MEPamp = mean(N_MEPamp);
std_MEPamp = std(N_MEPamp);
[numobs, numvars] = size(N_MEPamp);
STD = (N_MEPamp - repmat(mean_MEPamp,numobs,1)) ./
repmat(std_MEPamp,numobs,1);
for ax = 1:numvars
    ylim([-5, 5]);
    upperlimits = [4, 3, -2, -3];
    lowerlimits = [3, 2, -3, -4];
    colors = {'r', 'y', 'y', 'r'};
    for i = 1:4
        patch([0, 0, numobs, numobs],...
            [upperlimits(i), lowerlimits(i), lowerlimits(i),
upperlimits(i)],...
            colors{i}, 'EdgeColor', 'none', 'FaceAlpha', 0.4);
    end
end

offset = 4;

[X,Y] = pol2cart(...
    (0:1/numvars:1)*2*pi,...
    [STD(1,:), STD(1,1)] + offset);
plot(X,Y);
```

```matlab
axis off square;
hold on;

% Plot upper and lower control limits at 2 and 3 sd

for i = 1:4
    [X,Y] = pol2cart(...
        [0:1/99:1,1:-1/99:0]*2*pi,...
        [ones(1,100)* (upperlimits(i)+offset),...
        ones(1,100)* (lowerlimits(i)+offset)]);
    patch(X, Y, colors{i}, 'EdgeColor', 'none', 'FaceAlpha', 0.4);
end

% Label variables nicely

[X,Y] = pol2cart(...
    (0:1/numvars:1)*2*pi,...
    ones(1,numvars+1)*(4 + offset));
for i = 1:numvars
    h = text(X(i), Y(i), muscle{i});
    circlepos = (i-1)/numvars;
    if circlepos < 0.25
        set(h, 'VerticalAlignment', 'bottom', 'HorizontalAlignment',
'left');
    elseif circlepos == 0.25
        set(h, 'VerticalAlignment', 'bottom', 'HorizontalAlignment',
'center');
    elseif circlepos < 0.5
        set(h, 'VerticalAlignment', 'bottom', 'HorizontalAlignment',
'right');
    elseif circlepos == 0.5
        set(h, 'VerticalAlignment', 'middle', 'HorizontalAlignment',
'right');
    elseif circlepos < 0.75
        set(h, 'VerticalAlignment', 'top', 'HorizontalAlignment',
'right');
    elseif circlepos == 0.75
        set(h, 'VerticalAlignment', 'top', 'HorizontalAlignment',
'center');
    elseif circlepos < 1
        set(h, 'VerticalAlignment', 'top', 'HorizontalAlignment',
'left');
    end
end


title(Sign_Name)
end
```

subtitle.m: subtitle

```matlab
function [ax,h]=subtitle(text)
%
%Centers a title over a group of subplots.
%Returns a handle to the title and the handle to an axis.
% [ax,h]=subtitle(text)
%           returns handles to both the axis and the title.
% ax=subtitle(text)
%           returns a handle to the axis only.
ax=axes('Units','Normal','Position',[.075 .075 .85 .85],'Visible','off');
set(get(ax,'Title'),'Visible','on')
title(text);
if (nargout < 2)
    return
end
h=get(ax,'Title');
```

REFERENCES

Ajiboye, A. B., & Weir, R. F. (2009). Muscle synergies as a predictive framework for the EMG patterns of new hand postures. *Journal of neural engineering*, *6*(3), 036004.

Aoyagi, Y., Stein, R. B., Mushahwar, V. K., & Prochazka, A. (2004). The role of neuromuscular properties in determining the end-point of a movement. Neural Systems and Rehabilitation Engineering, IEEE Transactions on, 12(1), 12-23.

Barker, A. T., Jalinous, R., & Freeston, I. L. (1985). Non-invasive magnetic stimulation of human motor cortex. *The Lancet*, *325*(8437), 1106-1107.

Bernstein N. (1967). The Coordination and Regulation of Movements. Pergamon Press. New York.OCLC 301528509

Berry, M. W., Browne, M., Langville, A. N., Pauca, V. P., & Plemmons, R. J. (2007). Algorithms and applications for approximate nonnegative matrix factorization. *Computational statistics & data analysis*, *52*(1), 155-173.

Bizzi, E., Mussa-Ivaldi, F. A., & Giszter, S. (1991). Computations underlying the execution of movement: a biological perspective. *Science*, *253*(5017), 287-291.

Cheung, V. C., Piron, L., Agostini, M., Silvoni, S., Turolla, A., & Bizzi, E. (2009). Stability of muscle synergies for voluntary actions after cortical stroke in humans. *Proceedings of the National Academy of Sciences*, *106*(46), 19563-19568.

Cheung, V. C., Turolla, A., Agostini, M., Silvoni, S., Bennis, C., Kasi, P., ... & Bizzi, E. (2012). Muscle synergy patterns as physiological markers of motor cortical damage. *Proceedings of the National Academy of Sciences*,*109*(36), 14652-14656.

d'Avella, A., Saltiel, P., & Bizzi, E. (2003). Combinations of muscle synergies in the construction of a natural motor behavior. *Nature neuroscience*, *6*(3), 300-308.

Hallett, M. (2000). Transcranial magnetic stimulation and the human brain.*Nature*, *406*(6792), 147-150.

Huntley, G. W., & Jones, E. G. (1991). Relationship of intrinsic connections to forelimb movement representations in monkey motor cortex: a correlative anatomic and physiological study. *Journal of neurophysiology*, *66*(2), 390-413.

Giszter, S. F., Mussa-Ivaldi, F. A., & Bizzi, E. (1993). Convergent force fields organized in the frog's spinal cord. *The journal of neuroscience*, *13*(2), 467-491.

Kutch, J. J., Kuo, A. D., Bloch, A. M., & Rymer, W. Z. (2008). Endpoint force fluctuations reveal flexible rather than synergistic patterns of muscle cooperation. *Journal of neurophysiology*, *100*(5), 2455-2471.

Kutch, J. J., Kuo, A. D., Bloch, A. M., & Rymer, W. Z. (2008). Endpoint force fluctuations reveal flexible rather than synergistic patterns of muscle cooperation. *Journal of neurophysiology*, *100*(5), 2455-2471.

Lang, C. E., & Schieber, M. H. (2004). Human finger independence: limitations due to passive mechanical coupling versus active neuromuscular control.*Journal of neurophysiology*, *92*(5), 2802-2810.

Lee, D. D., & Seung, H. S. (2001). Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems* (pp. 556-562).

Loeb, E. P., Giszter, S. F., Saltiel, P., Bizzi, E., & Mussa-Ivaldi, F. A. (2000). Output units of motor behavior: an experimental and modeling study. *Journal of Cognitive Neuroscience*, *12*(1), 78-97.

Mussa-Ivaldi, F. A., Giszter, S. F., & Bizzi, E. (1994). Linear combinations of primitives in vertebrate motor control. *Proceedings of the National Academy of Sciences*, *91*(16), 7534-7538.

Mushahwar, V. K., Aoyagi, Y., Stein, R. B., & Prochazka, A. (2004). Movements generated by intraspinal microstimulation in the intermediate gray matter of the anesthetized, decerebrate, and spinal cat. *Canadian journal of physiology and pharmacology*, *82*(8-9), 702-714.

Paatero, P., & Tapper, U. (1994). Positive matrix factorization: A non‐negative factor model with optimal utilization of error estimates of data values.*Environmetrics*, *5*(2), 111-126.

Rose, P. K., & Scott, S. H. (2003). Sensory-motor control: a long-awaited behavioral correlate of presynaptic inhibition. *Nature neuroscience*, *6*(12), 1243-1245.

Rothwell, J. C. (1997). Techniques and mechanisms of action of transcranial stimulation of the human motor cortex. *Journal of neuroscience methods*, *74*(2), 113-122.

Ting, L. H., & Macpherson, J. M. (2005). A limited set of muscle synergies for force control during a postural task. *Journal of neurophysiology*, *93*(1), 609-613.

Tresch, M. C., & Bizzi, E. (1999). Responses to spinal microstimulation in the chronically spinalized rat and their relationship to spinal systems activated by low threshold cutaneous stimulation. *Experimental brain research*, *129*(3), 401-416.

Tresch, M. C., Cheung, V. C., & 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.

Tresch, M. C., & Jarc, A. (2009). The case for and against muscle synergies.*Current opinion in neurobiology*, *19*(6), 601-607.

Turvey, M. T. (2007). Action and perception at the level of synergies. *Human movement science*, *26*(4), 657-697.

Valero-Cuevas, F. J., Venkadesan, M., & Todorov, E. (2009). Structured variability of muscle activations supports the minimal intervention principle of motor control. *Journal of neurophysiology*, *102*(1), 59-68.