EMG-based Robot Control Interfaces: Beyond Decoding

by

Chris Wilson Antuvan

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved April 2013 by the
Graduate Supervisory Committee:

Panagiotis Artemiadis, Chair
Jitendran Muthuswamy
Veronica J. Santos

ARIZONA STATE UNIVERSITY

May 2013

ABSTRACT

Electromyogram (EMG)-based control interfaces are increasingly used in robot teleoperation, prosthetic devices control and also in controlling robotic exoskeletons. Over the last two decades researchers have come up with a plethora of decoding functions to map myoelectric signals to robot motions. However, this requires a lot of training and validation data sets, while the parameters of the decoding function are specific for each subject. In this thesis we propose a new methodology that doesn't require training and is not user-specific. The main idea is to supplement the decoding functional error with the human ability to learn inverse model of an arbitrary mapping function. We have shown that the subjects gradually learned the control strategy and their learning rates improved. We also worked on identifying an optimized control scheme that would be even more effective and easy to learn for the subjects. Optimization was done by taking into account that muscles act in synergies while performing a motion task. The low-dimensional representation of the neural activity was used to control a two-dimensional task. Results showed that in the case of reduced dimensionality mapping, the subjects were able to learn to control the device in a slower pace, however they were able to reach and retain the same level of controllability. To summarize, we were able to build an EMG-based controller for robot devices that would work for any subject, without any training or decoding function, suggesting human-embedded controllers for robotic devices.

ACKNOWLEDGEMENTS

It has been a great experience for me to study and do some valuable research at Arizona State University. It's a dream come true. I would like to thank my Dad, Mom, brother and sister for supporting and encouraging me to pursue my masters degree. I am indeed grateful to Dr. Panagiotis Artemiadis for taking me in his lab and giving me the opportunity to do some great research in the field that I am really excited about. I thank you for all the constructive criticism and encouragement during both failure and success. This thesis would not have been possible without you. I would also like to thank my relatives and friends for their affection and efforts in motivating me. A special mention to all my lab mates for helping me in experimentation and in resolving problems. Above all, I would like to dedicate this small achievement to God, who is the reason for what I am now. I hope whatever research I have done would be useful to mankind.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

Human Machine Interfaces have started getting more popular among the robotics research community since the last two decades, with a great deal of work involving applications like neuro-prosthetics, orthotics, teleoperation and other areas. With the advent of technologies which can elucidate useful information from the brain and muscles like Electroencephalography (EEG) and Electromyography (EMG), researchers have stretched its application by trying to effectively establish a connection between electromechanical systems and humans. Numerous studies and experiments have been conducted in order to utilize this technology, in order to help amputees regain to some extent, the lost ability to perform daily life tasks using active prostheses. Moreover, exoskeletons (orthoses) have been developed to leverage or augment humans in performing tasks beyond their physical capacities, or to enable long distance teleoperation of robotic devices [29, 30, 11, 12, 3].

The main challenge in this field lies in decoding the brain or muscle signals to motion. A plethora of decoding algorithms have been developed to map EEG or EMG signals to useful control outputs, necessary to operate robotic devices. Most of these techniques are subject-specific and involve pre-processing of a training data set, in order to train a decoder and apply it for the real-time control of the device. Some of these decoding algorithms work only for specific gestures or motions of the hand. The accuracy of the decoded motion however is limited, since it is based on a system trained with a finite set of training data, preventing generalization to unseen signals.

This thesis mainly aims at building a robust EMG-based decoder for the real-time control of robots. Instead of trying to train the *perfect* decoder, the main idea is to supplement the decoder performance by making the user learn the inverse model of the decoder and make up for the decrease in efficiency. This comes up from the fact that human beings are capable of learning a wide variety of tasks, which is evident in their daily life activities.

We also further optimized the algorithm taking into account that muscles develop synergies while performing a task. This approach is to reduce the dimensionality of control and to optimize the learning rate for the control of any device. Learning patterns are analyzed in both the cases to understand the extent and retainment of learning to control an external device using neural signals.

1.1 Motivation

A great amount of research in this field has been dedicated in building new and better algorithms to decode myoelectric and encephalographic signals to motions of the human body. But the main challenge is attributed to the fact that the system that is to be modeled is highly non-linear and it is very difficult to achieve good decoding accuracy [31]. The results of the experiments done by Cunningham et al[9] and Taylor et al[28] show that the performance of the decoding algorithms turns out to be better in the context of a closed loop system. It suggests that humans are capable of modifying their neural activity in order to match the desired trajectory, due to the presence of feedback. In most of the cases there is always the constraint of user specificity, requiring to change the parameters of the decoder for each subject. This means that each time a new subject has to do the experiment and also possibly the same subject during different sessions, a separate training data set has to be collected and processed in order to modify the parameters of the decoder. The initial decoder training phase involves a great deal of time for acquiring the data and then processing it. This is not practical for real-life application e.g. active prostheses or exoskeletons. A more intuitive and easy to use interface is needed.

Gage et al[14] designed their experiments based on a non biomimetic type decoder where naive animals who didn't receive any prior training, learned to modify their neural activity in order to successfully perform the task. The animals were provided with biofeedback. The results showed that the brain signals and the corresponding myoelectric signals which were generally co-activated could be altered. Recovering the original motor function by an alternate neural pathway explains the flexibility of brain to adjust to different neural

mapping. It further explains that the plasticity of the motor cortex definitely does play a great role in successful brain and/or myoelectric controlled interfaces. These ideas shift the focus towards an alternate and better strategic control solution for brain and muscle signal decoding. Thereby, instead of trying to improve existing algorithms or coming up with new algorithms to achieve better open loop offline performance, we need to utilize the human brain's capability to modify the neural activity and make up for the loss in decoding efficiency. In other words, we need to enable the subject learn the inverse model of a decoder.

The human brain is capable of learning and retaining a large number of complicated tasks. Learning is defined as *acquiring new, or modifying existing, knowledge, behaviors, skills, values, or preferences and may involve synthesizing different types of information*. The ability to learn is a characteristic of most living organisms. Learning doesn't happen instantaneously but rather, is a process, and varies with different individuals. Once learned the skills are relatively permanent[27]. The hypothesis being tested here is whether the human brain is capable of learning any arbitrary mapping of signals to action, and how this could be exploited in building a robust myoelectric-based robot controller. Also we propose that the mapping does not have to be intuitive for the subjects to learn.

In the case of Brain Computer Interfaces, electrical activity of the brain due to the ionic flow in the neurons, referred to as EEG signals, are measured by placing electrodes on the motor cortex area, which is then decoded to generate the required control movements. The main drawbacks of using Electroencephalography is that it has low spatial resolution, poor detection of neural activity below the upper layers of the brain, and also poor signal to noise ratio. Better and much clear signal can be obtained by using invasive electrodes.

EMG signals can be considered as a useful proxy to the EEG signals of the brain. The brain sends signals to stimulate respective muscles in order to produce motion of the joint. Myoelectric Controlled Interfaces (MCI) are also largely used since they are non-

invasive. Use of EMG's is quite practical in the control of powered prosthetics and exoskeletons. EMG signals promote a much better understanding of the intended motion, since they are only generated when the muscle is stimulated which directly implies the intended motion. Also EMG signals can describe co-contraction, i.e. adjustment of musculoskeletal impedance.

1.2 Objective

The main objective of this thesis is to develop a robust EMG-based control strategy for robotic devices, which does not require any offline processing of data in order to train the decoder. An arbitrary non-intuitive mapping function is fixed between muscle signals and the device motion. The subjects do not have prior knowledge about the controls necessary to accomplish the task. The only way for the subjects to successfully reach the targets would be to learn the inverse model of the mapping function. There definitely would be a learning curve associated with mastering the task, but it is really important to analyze whether the skill set is retained or not. Brain develops a synergistic muscle activation patterns when they learn a task. Taking that into consideration, it is possible to optimize the control algorithm by using a reduced dimensionality of operation which could be easier for the subjects to learn. Dimensionality reduction is examined using Principal Component Analysis, and the same task is performed using a mapping function in the low dimensional space. Both cases are compared in terms of learning rate and skill retainment.

Chapter 2

Background Literature

2.1 Introduction

The use of electromyographic signals to control prosthetic devices has been researched since early 1970's[16]. However, due to the lack of advanced computing facilities and bulky hardware for acquiring data, myoelectric based control for real time operation was not feasible at that time. Extensive research is going on in this field to be able to effectively utilize brain and muscle signals to control prosthetic and orthotic devices.

The human arm is a highly complicated system with more than thirty muscles, controlling around twenty one degrees of freedom. Many researchers have come up with pattern recognition algorithms that classify certain arm or hand motions and gestures based on EMG signals. The first of its kind can be found in [20]. It involved a basic time domain statistics and a multilayer perceptron neural network classifier which was capable of classifying four kinds of motion with reasonable accuracy. Many researchers have come up with different kinds of classification techniques like the Gaussian Mixture Models, Artificial Neural Networks, Support Vector Machine, Kalman Filters etc. There are also continuous decoding models like the Black Box Modeling techniques. Across all these methods, a training data set is collected, consisting of the subject's myographic signals and other kinematic or gesture information. The data needs to be processed in order to identify the decoding function and is then validated using another set of data, to find out how accurately the decoder interpreted the signal.

Electromyography

Electromyography (EMG) is an experimental technique concerned with the development, recording and analysis of myoelectric signals. Myoelectric signals are formed by physiological variations in the state of muscle fiber membranes [22]. Besides physiological and biomechanical studies, electromyography has found applications in various fields of

research like rehabilitation, sports science and ergonomics as shown in Fig.2.1. The amplitude of the EMG signal after rectification and smoothening is related to the force or torque exerted by the subject. However, there are a lot of factors which influence the EMG signal like physiological, anatomical, as well as technical aspects that have to do with its recording. It is not possible to regulate all these factors with the current technology[23]. The factors that influence the EMG signal can be broadly classified into three categories:

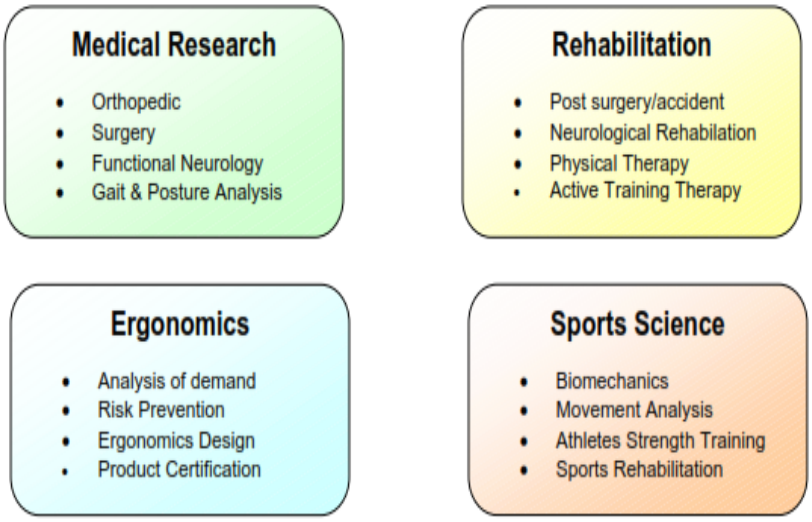

Figure 2.1: Applications of electromyography [22].

1. *Causative:* Causative factors are those which have the elemental effect on the signal. This is further divided into *Extrinsic* and *Intrinsic*. Extrinsic factors include 1) The electrode configuration 2) location of the electrode with respect to the motor points on the muscle 3) location of the electrode on the surface of the muscle and 4) orientation of the detection surfaces with respect to the muscle fibers. The Intrinsic factors include 1) number of active motor units at any particular time of the contraction 2) fiber type composition of the muscle 3) blood flow in the muscle 4) fiber diameter 5) depth and location of the active fibers within the muscle with respect to the electrode detection surfaces 6) the amount of tissue between the surface of the muscle and the

electrode and 7) other factors that are yet to be identified, such as the length of the depolarization zone and ionic fluxes across the membrane, etc.

2. *Intermediate:* Intermediate factors are those physical and physiological phenomena which are influenced by one or more of the *Causative* factors, and also influence the *Deterministic* factors. They include 1) the band-pass filtering aspects of the electrode 2) the detection volume of the electrode 3) superposition of action potentials in the detected EMG signal 4) crosstalk from nearby muscles 5) the conduction velocity of the action potentials that propagate along the muscle fiber membrane and 6) the spatial filtering effect due to the relative position of the electrode and the active muscle fibers.

3. *Deterministic:* Deterministic factors are those which have a direct relationship with the EMG signal and the recorded force. They include 1) the number of active motor units 2) the motor unit force-twitch 3) the mechanical interaction between muscle fibers 4) the motor unit firing rate 5) the number of detected motor units the amplitude, duration and shape of the motor unit action potentials and 6) the recruitment stability of motor units.

2.2 Classification techniques

Gaussian Mixture Model

The Gaussian Mixture Density is found by taking the weighted sum of M component densities [19]. In case of an N-class pattern recognition system there could be N Gaussian mixture models (GMM) associated with each class. For a D-dimensional feature vector denoted as $\vec{x}$, the mixture density for the nth model is defined as

$$p(\vec{x}|\lambda_n) = \sum_{i=1}^{M} \omega_i^n p_i^n(\vec{x}) \tag{2.1}$$

where M is the number of mixture components; ω_i^n, i=1,...,M, are the mixture weights that satisfy the constraint $\sum_{i=1}^{M} \omega_i^n = 1$ and $\omega_i^n \geq 0$. The weighted linear combination of

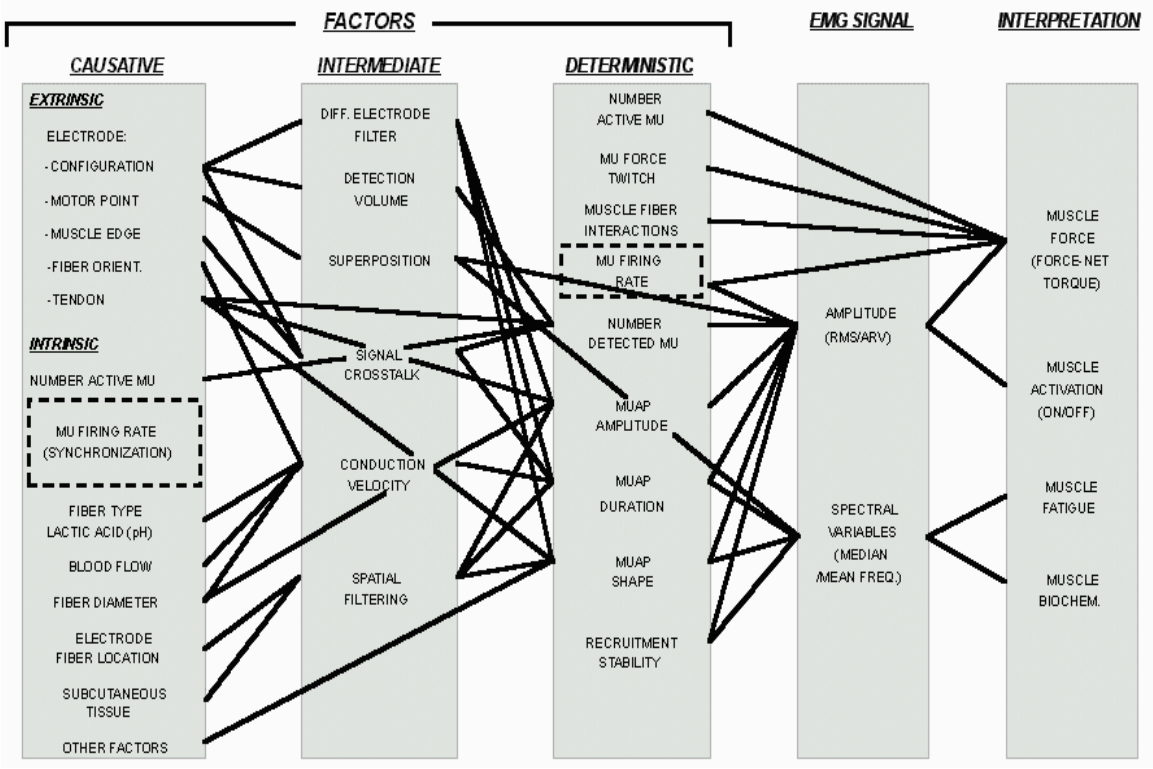

Figure 2.2: A schematic diagram of the factors which affect the EMG signal [23].

M component uni model Gaussian density functions give the mixture density, $p_i^n(\vec{x}), i = 1,...,M$. Each component density is a D-variate Gaussian function parameterized by a $D \times 1$ mean vector, $\vec{\mu_i^n}$ and $D \times D$ covariance matrix, C_i^n.

$$p_i^n(\vec{x}) = \frac{1}{(2\pi)^{D/2}\left|C_i^n\right|^{1/2}} exp\left\{-\frac{1}{2}(\vec{x}-\vec{\mu_i^n})'(C_i^n)^{-1}(\vec{x}-\vec{\mu_i^n})\right\}. \tag{2.2}$$

The mixture weights, mean vectors and covariance matrices parameterize the complete Gaussian mixture density which is denoted as $\lambda_n = \omega_i^n, \vec{\mu_i^n}, C_i^n$ The main goal of the GMM-based training is to estimate the parameters λ_n, so that the Gaussian mixture density can match the distribution of the training feature vectors. The parameters of the Gaussian Mixture Model are calculated using the Expectation Maximization algorithm which does iterations to maximize the likelihood generated by each GMM, $p(\vec{x}|\lambda_n)$, based on the data for a particular subject. Validation is done using the test data, in order to find the class

model which has maximum a posterior probability for the given observation details. Selection of the optimum is very crucial when dealing with EMG-based classification in GMM. Too few number of mixture components would reduce the accuracy of the model, and too large a number of mixture components would reduce the performance of the model. Also a variance limiting constraint is necessary to avoid the singularity of the likelihood model when dealing with a large number of mixtures.

Artificial Neural Networks

These are algorithms which were aimed at modeling the human brain. In this method, a large parallel layers of neurons or computational units are formed. During the learning procedure or training phase, the weights of the neurons are modified so as to match the required output. It uses back propagation algorithm to reduce the error between the generated and desired output. The error is then propagated to the next level of neurons and it repeats till the error is considerably reduced [20].Among the main advantages of Neural Networks include its parallel distributed structure and its ability to generalize [17]. The input of the neuron j is modeled as

$$v_j(n) = \sum_{i=0}^{m} \omega_{ji}(n) y_i(n) \tag{2.3}$$

where $y_i(n)$ is the input to neuron j and $\omega_{ji}(n)$ is the weighting between the input and neuron j. The drawback of this method is that there is no assurance that the error converges to a global minimum. Also the training process is long and it might not produce good results outside the training data set.

Support Vector Machine

The concept of Support Vector Machine (SVM) was proposed by Vapnik in 1992 [4]. The classification strategy is of binary model, which decides whether a particular data belongs to a set of category or not. Multiclass SVM is required to classify data into more than two categories. The optimization between complexity of the model and the empirical error is

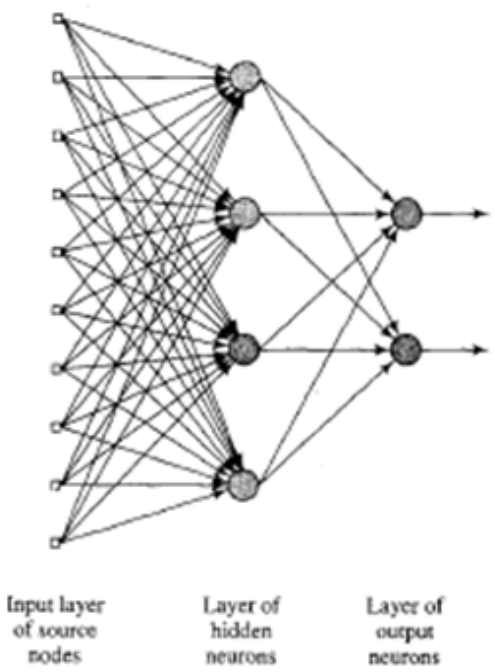

Figure 2.3: Neural Network with one hidden layer and one output layer[20]

done by making use of the concept of structural risk minimization. It involves building a decision surface, which is basically a hyperplane for a general input vector of the form R^n. If we consider a simpler case where a line is used as a hyperplane to separate data in a two dimensional space. The optimal hyperplane is given by

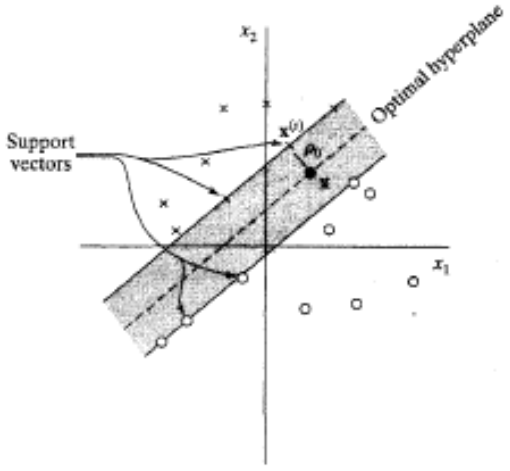

Figure 2.4: Optimal hyperplane with linearly separable data [7]

$$w^T x + b = 0 \tag{2.4}$$

where w is called the weight vector and x the input vector. The input vectors which define the hyperplane are called support vectors. The necessary conditions for classification are

$$\begin{aligned} w^T x_i + b \geq 0 \qquad & y_i = +1 \\ w^T x_i + b < 0 \qquad & y_i = -1 \end{aligned} \tag{2.5}$$

where $\mathbf{x}_i$ is the i^{th} sample and y_i is the i^{th} output. The drawbacks of this method arise when data lies within the margin of separation which makes the data inseparable. Solutions to this problem has been proposed. One unique advantage of SVM is their ability to generalize. By utilizing the deviation of input vectors as a variable in the algorithm for optimization, SVMs can account for some variability of the input vector which makes it a good candidate for classifying EMG signals[7].

2.3 Continuous Decoding Techniques

System Identification

System Identification or Black Box Modeling is a method of identifying the system structure which would map any input signals to the output produced. These algorithms are mostly used when it is difficult to model the structure or dynamics of a system. So in that case the inputs to the system and the corresponding outputs are used to fit a model of the system. There are different kinds of black box models that are used to identify the structure. There is the Linear Auto Regressive Model (ARM) like the ARX, ARMAX, Box-Jenkins, as well as the State Space Model. Many researchers have tried to use black box models to map myoelectric signals to kinematics of the arm. MATLAB has a System Identification toolbox, which would build the decoding function and also identify the accuracy of the decoder, when the training and validation data sets are entered. Comparison could also be done across various types of models to identify the model which gives better accuracy. In the work by Artemiadis et al [1], an ARMAX model was used to model the function that took as inputs EMG signals from two muscules, the Biceps Brachii and the Triceps

Brachii, and generated an output which was the elbow joint angle. Fig.2.5 shows an ARMAX model in the discrete space. u_k represents the input to the system, y_k the output, e_k the model disturbance, A, B and C the polynomial functions of the model.

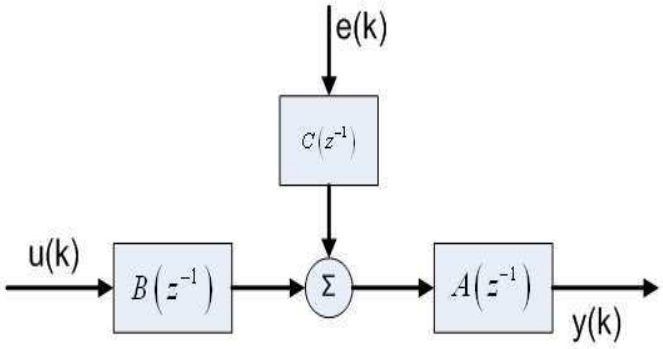

Figure 2.5: ARMAX model [1].

2.4 Related Work

Learning Inverse Model

There has been some work in simulating how human brain learns the inverse model of a particular decoding function. In the work by Carmena [18], a model of the brain was assumed, which takes in as input the firing properties of a certain set of neurons and outputs the x and y position of an end effector. The experiment involves the commonly used center to reach out tasks, where the aim is to reach one of the eight peripheral targets on a circle when instructed. The algorithm consists of the decoding function, an inverse model and a feedback controller, as shown below:

$$Y^{xy} = D(f) \tag{2.6}$$

where f is the vector containing the firing rates of the recorded cells, D is the decoder function and Y^{xy} is the position vector of a two-dimensional cursor. The brain does the encoding to modify the input signal based on the relation

$$f = B(T^{xy}, Y^{xy}, \lambda) \tag{2.7}$$

where T^{xy} is the $x-y$ position of the target to be reached and λ is a set of parameters that the brain adapts for learning the inverse model. The error between the intended desired and actual position is fed to the feedback controller, and based on that it modifies the inverse model, in order to reduce the error. The block diagram of the algorithm used is as shown in Fig.2.6.The inverse model is updated based on the corrections made by the feedback controller. The authors used non gradient based techniques in order to modify the inverse model based on the error. This is an analogy to how the brain modifies the neural signals based on the output generated in order to develop the inverse model of the decoding function. In Fig.2.7 blue line is the target evolution over time (X and Y axis plotted separately); red dotted line is the actual trajectory followed by the simulated BMI system. Top of the bold line is before learning; bottom is after learning.

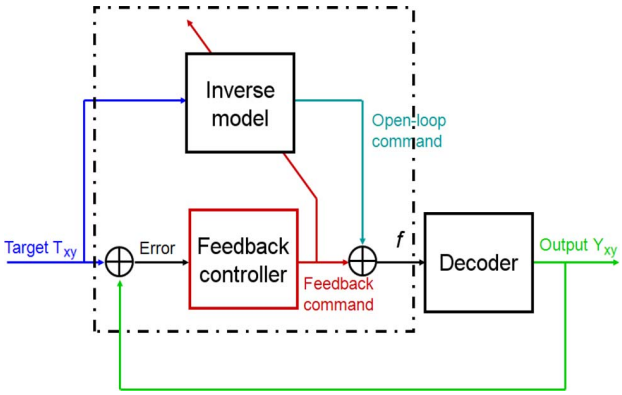

Figure 2.6: Feedback-error learning model [18].

Another study by Gage et al[14] showed similar results. They investigated how naive subjects who did not have any prior training to control a device using the brain cells of the cortex, gradually learned to control the device. They used six untrained rats to perform control of an auditory device. The experiment consists of an audio cursor represented by 90ms sound pips representing the one dimensional location within the logarithmically spaced 250 Hz to 16 kHz frequency spectrum. Baseline firing rates were mapped to the center of the frequency space, and in each trial a specific target tone at a particular frequency

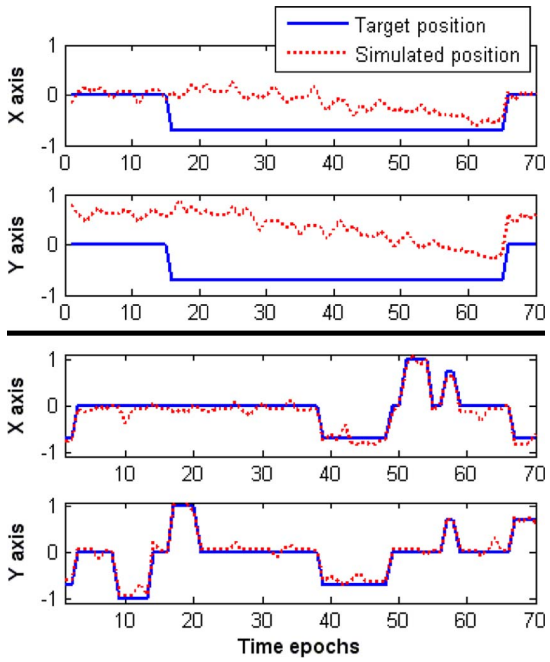

Figure 2.7: Effect of learning on the reaching trajectories [18].

is presented. Subjects had to match the tone by making use of their cortical neurons which was decoded in real time. Kalman filtering was used to decode the frequency of the cursor. Fig.2.8 shows the results from the experiment. The top line shows the mean of the group's performance. Error bars represent standard error. Gray band indicates the 95% confidence interval. Bottom lines indicate the length of time each subject was performing the task. Tick marks indicate early learning sessions. Days of training and mean percent correct for all subjects for each day are indicated across the top.

Importance of Feedback in Closed Loop Control

In the work by Taylor et al[28], the authors investigated the control of a 3D cursor in monkeys by using the cortical neurons. In the paper they compared cell tuning properties for the same trajectories in both brain control and hand controlled movements. Hand controlled motion involved an open loop model, where the monkey performed the task and the corresponding brain activity was recorded offline. In the brain controlled task, visual feedback of a virtual cursor was provided as the monkey performed the same task. There was

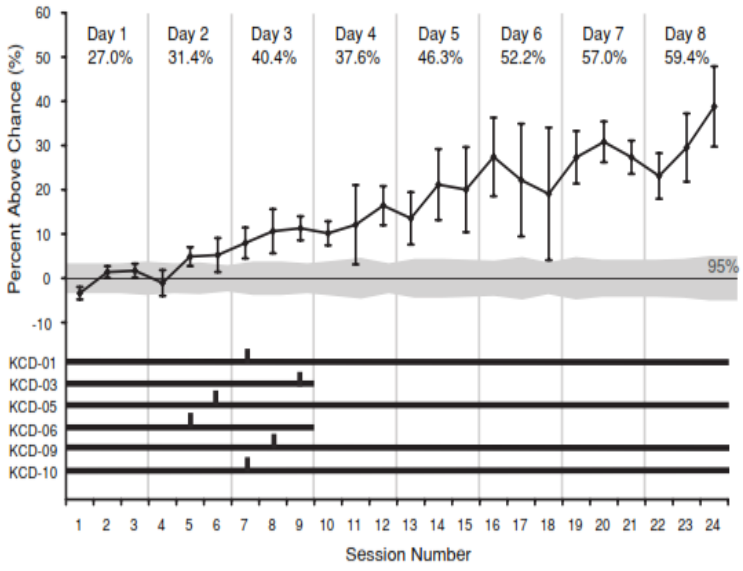

Figure 2.8: Performance in the fixed target task for six rats across 8 days [14].

a significant change in the brain cell firing patterns for performing the same task in brain controlled and hand controlled experiment. After around 30 days of training in showed that there was great control of the cursor in terms of timing and direction in the brain controlled experiment when compared to the hand controlled experiment. Thus it showed the ability of the brain to co-adapt to the reaching motion by modifying its firing properties. It also showed how feedback could aid in better control of the task. Fig.2.9 shows the table containing results in both the closed loop and open loop brain controlled trajectories. It shows better performance in closed loop system.

There has been a smaller number of studies that have demonstrated online closed loop control and showed that prostheses can be meaningfully controlled by monkeys or humans (Carmena et al 2003[5]; Chase et al 2009[6]; Ganguly and Carmena 2009[15]; Kim et al 2008[25]; Koyama et al 2010[26]; Santhanam et al 2006[13]). In the work by Kim et al[25], the authors have compared the performance of a decoding algorithm in both open loop and closed loop and it shows the importance of closed loop training for better performance of the decoder. In the work by Chase et al[6] the authors have compared the performance of two decoding algorithms in both offline and online based control strategies.

	Monkey		
	L	M	Both
Closed-loop brain-controlled trajectories			
% Targets hit	52 ± 14	46 ± 18	49 ± 17
% Time in correct octant	36 ± 9	34 ± 11	35 ± 11
Open-loop brain-predicted trajectories			
% Targets hit	32 ± 11	23 ± 5	27 ± 9
% Time in correct octant	23 ± 9	23 ± 9	23 ± 9
Miscellaneous			
Cells used	18 ± 4	18 ± 3	18 ± 4
Mean R^2	0.63 ± 0.07	0.64 ± 0.09	0.64 ± 0.08

Figure 2.9: Mean and standard deviation of daily statistics from the open versus closed-loop experiment [28].

These algorithms which showed stark performance difference in offline training, had less performance difference in a closed loop control with feedback, showing the impact of feedback in closed loop control. They have all iterated the need for the presence of feedback for successful implementation of brain machine interfaces. Koyama et al[26] also further did comparisons between offline and online performances based on their directional basis and its implications. Cunningham et al[9], developed an online prostheses simulator to closely model the human feedback control system, where synthetic neural signals are used in the experiment instead of using monkeys or humans. As shown in Fig.2.10 the x-axis shows 4 testing paradigms in terms of increasing realism. Offline data analysis is perhaps the least reasonable proxy to eventual user mode, as it entirely neglects the closed-loop control. On the other end of the spectrum is the human clinical trial, which is precisely the eventual user mode. Left axis (blue) shows the difficulty associated with testing each algorithm or algorithmic parameter setting. Right axis (red) shows the number of algorithm and parameter choices that are reasonably testable, given costs and other constraints.

The use of closed-loop myoelectric controlled interfaces was investigated by Radhakrishnan et al[24], where the authors were interested in understanding human learning. The experiment consisted of two types which they classified as intuitive and non intuitive

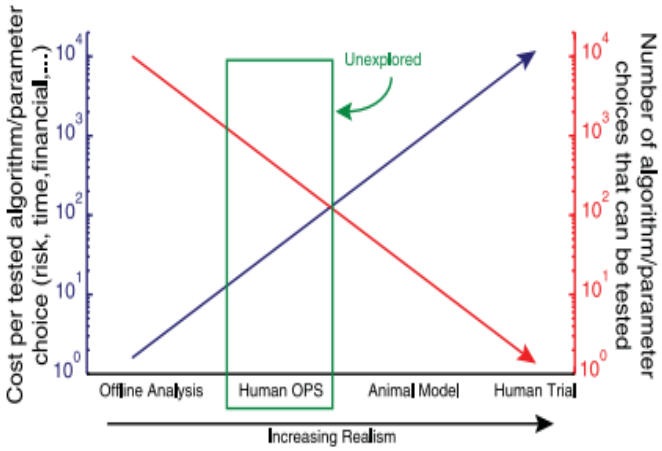

Figure 2.10: Concept figure for online prosthetic simulator (OPS) opportunity[9]

type mapping, where EMG signals from eight muscles (Biceps, Triceps, Deltoid, Flexor Carpi Ulnaris, Extensor carpi radialis, First dorsal interosseus, Abductor pollicis brevis, Abductor digiti minimi, Abductor pollicis longus) were decoded to generate the two dimensional cursor position. In intuitive mapping, six out of eight muscles was used to produce the motion of cursor along particular direction on the two dimensional plane. These directions were consistent with the subject's action on the limb in that posture. Cursor movement was determined by adding the six vectors. Whereas in non intuitive type mapping, the six out of the eight muscles were randomly represented along the equally spaced direction of action. The direction along which the muscle produced motion of the cursor in both the types of experiment is as shown in Fig.**??**. Subjects were able to learn the mapping in both the intuitive and non-intuitive experiments, but the learning time was more in the case of non-intuitive mapping. The next set of experiments were to see if proprioceptive feedback does play a role in learning. For that they introduced vibrations using a linear motor at a frequency of 50 Hz. The main aim was to introduce noise into the sensory feedback. Then they compared the performance of the subjects in the experiments with and without vibration. The plots show a higher learning period in the experiment with vibrations. It clearly shows the importance of non visual feedback in learning myoelectric mapping. In

Fig.2.12, A shows the trajectories of the cursor center to each of the 12 targets before and after learning the non-intuitive MCI arrangement. Small circles indicate position of cursor 500 ms after the target appeared. B: Movement time for each individual trial plotted for a single subject learning the non-intuitive arrangement. C: Mean movement time over blocks of 12 consecutive trials averaged across all subjects for intuitive and non-intuitive MCI arrangements. Bars indicate standard error of mean.

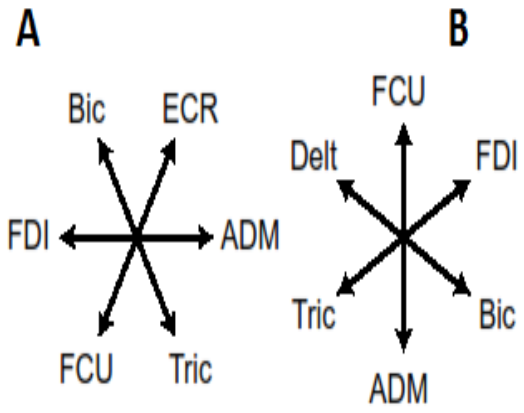

Figure 2.11: A: shows arrangement of muscles for intuitive mapping, B: shows arrangement for non intuitive mapping [24].

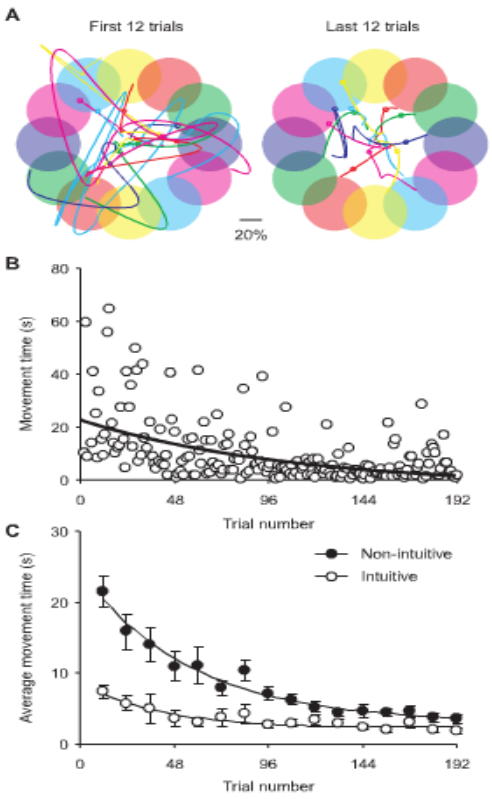

Figure 2.12: Motor learning during Myoelectric Control Interface(MCI) operation [24].

Muscle Synergies

The Central Nervous System (CNS) has to coordinate the various muscles of the body to generate the required motion of the body. The large number of degrees of freedom of the musculoskeletal system makes it a highly non linear and complicated mapping between muscle activity and movement. In such a high dimensional and complex system the brain cannot rely on feedback from the sensory system in order to produce movement. It is done by means of an open loop control and in certain coherent patterns of muscle activation. The CNS does any motion in a feed-forward manner, unless there arises an error in the planned motion, leading it to switch to a feedback control in order to correct the error and move towards the desired trajectory. Also there are certain patterns or combinations of muscle activations that are found when any motion task is performed. The main idea behind these types of muscle activations is to reduce the dimensionality of the control input. In that way the CNS doesn't have to spend too much time in controlling each muscle separately. Some actions such as locomotion which involves repetitive patterns are mediated directly from the spinal cord, which also generates synergistic activation of the muscles.

In the work by D'Avella et al[10], the authors have identified the muscle activation patterns of intact and unrestrained frogs during kicking, which is a natural defensive mechanism. They proposed that the muscle activation patterns were due to time varying muscle synergies which are a group of muscles having both amplitude and time-varying profiles. This small number of muscle synergies was responsible for producing the entire repertoire of movements. They also devised an algorithm to identify the muscle synergies from the muscle activation patterns that was recorded during the kicking of the frog which is as shown in Fig.2.13.

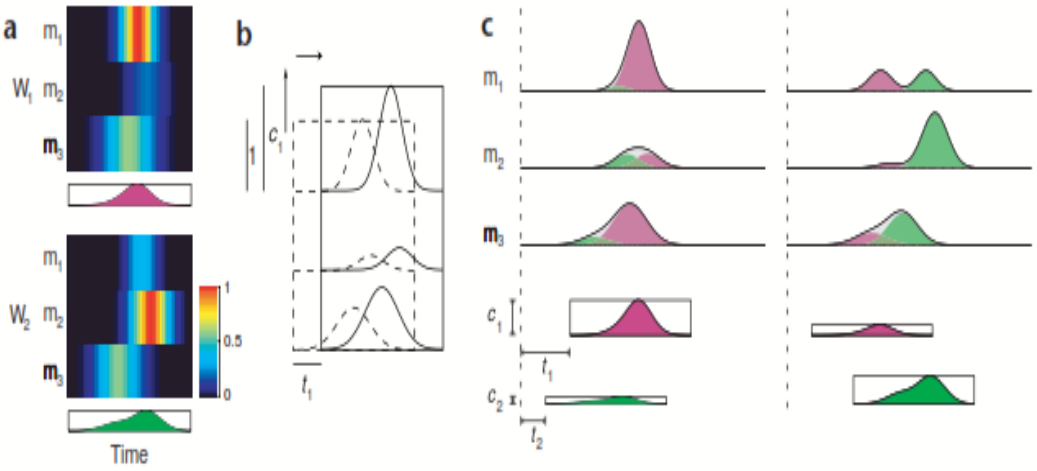

Figure 2.13: Time-varying synergies model [10].

Principal Component Analysis (PCA)

Dimensionality reduction comes into picture because it is difficult to visualize and process high-dimensional data. The idea behind PCA is that any sample dataset in an N dimensional space can be represented to some extent in a lower dimensional space. For detailed explanation on PCA the reader should refer to [21]. PCA leads to a new way of representing the N dimensional space, using a new set of axes defined by the principal component vectors i.e. the eigenvectors. The new set of coordinate axes provides a much better way of representing and analyzing the original data set. Many researchers have proved that there is high correlation in muscle activation patterns while performing a task. In the previous work by Dr. Artemiadis et al[2], the authors have used the PCA to control a robot in reduced dimensional space. Principal Component Analysis is a powerful tool in analyzing strong correlation between different dimensions too.

The main steps involved in finding the principal component vectors of EMG signals are listed below:

1. *Getting Data:* The data set consists of the EMG signals of the muscles from which

the synergistic effect needs to be identified. Each muscles represent independent variables in an N dimensional space, where N is the number of muscles.

2. *Subtract the mean:* The mean along each dimension or muscle is calculated and it is subtracted from the samples along the respective dimensions.

$$X_i^n = X_i^n - \bar{X}_i \tag{2.8}$$

where i represents the i^{th} muscle,n the n^{th} sample, X_i^n represents the i^{th} EMG signal at the n^{th} sample, while $\bar{X}_i$ represents the average of the EMG signal of the i^{th} muscle.

3. *Calculate the Covariance Matrix:* The covariance matrix C is given by

$$C_{ij} = cov(X_i, X_j) = E[(X_i - \mu_i)(X_j - \mu_j)] \tag{2.9}$$

where $\mu_i = E(X_i)$ is the expected value of the i^{th} muscle, i.e. its mean value.

4. *Calculate the Eigenvectors and Eigenvalues of the covariance matrix:* The covariance matrix is an $N \times N$ square matrix. The Eigenvectors and Eigenvalues of the covariance matrix give very useful information about the data that is to be analyzed. The Eigenvectors are of unit magnitude and are mutually perpendicular. The Eigenvectors point along directions where majority of the data is located. The Eigenvalues provide information about the significance of the corresponding Eigenvectors. The higher the Eigenvalue, the more significant is the corresponding Eigenvector in representing original data variability.

5. *Choosing the Components and forming the Feature Vector:* This is the step where dimensionality reduction comes into picture. The Eigenvalues tells us which Eigenvectors are more significant or those axis where majority of the data is described by. We need to find out those vectors from the entire set, such that it is still possible to represent the majority of the data using those vectors alone. For doing that, we need

to find out the *explained variance* which is defined by:

$$v_i = \frac{\lambda_i}{\sum_{j=1}^{N} \lambda_j} \tag{2.10}$$

where $i = 1, ..., N$ and N represents the dimension or the number of muscles, λ_i are the Eigenvalues of the covariance matrix C. The Explained Variance of an Eigenvector gives the percentage of the entire variance that could be represented using that Eigenvector. A *feature vector* doesn't have to contain all the Eigenvectors, if dimensionality reduction is required, however in the general case the feature vector is defined by:

$$V = [eig_1 eig_2 ... eig_N] \tag{2.11}$$

6. *Deriving the New Data Set:* A *feature vector* with all the Eigenvectors doesn't modify the data set. It only represents the same data set in a new frame of reference. Dimensionality reduction comes into picture when the *feature vector* contains only a subset of the most important Eigenvectors. The data set is therefore compressed or transformed to a new data set, described by the selected only Eigenvectors with a high Explained Variance, as shown below:

$$F = V^T G \tag{2.12}$$

where F is the reduced data set, an $n \times m$ matrix with n being the reduced dimension (varies from 1 to N) and m the number of samples, V is an $N \times n$ matrix with the eigenvectors selected, and G is an $N \times m$ matrix which is the original data set.

Chapter 3

Methodology

3.1 Introduction

The objective of this thesis is to investigate human learning of an abstract mapping function and also optimize the control algorithm for better learning. The experimental design will involve the mapping of EMG signals to the motion of a virtual object in 2D. The safest and easiest approach for a subject to generate controlled motion of an object would be through graphical interface rather than using actual hardware. We are going to use a cursor on a monitor as the virtual object, as this is the commonly used scenario in such studies. The motion of the cursor will be on a two dimensional plane. The minimum number of control inputs that would be required to control the motion in a two dimensional space is two. The two inputs are similar to the x-y coordinates in the Cartesian plane. However, there should be at least two muscles to control the motion in any particular axis, because we need both positive and negative motion. The best way to choose two muscles suitable for each motion axis would be if they are antagonistic, meaning they oppose each other. That is to say that when motion happens along one direction in any motion axis, it wouldn't be possible to produce motion in the opposite direction simultaneously, unless the subject co-contracts his/her muscle. So the pair of antagonistic muscles that was chosen, keeping in mind the ease of experimentation for subjects, includes the Biceps Brachii (BB) and Triceps Brachii (TB), the Flexor Carpi Ulnaris (FCU) and the Extensor Carpi Radialis (ECR).

3.2 Equipment Used

The experimental setup includes the *Delsys Trigno Wireless Electromyogram System*, a *National Instruments USB-6343 X Series Data Acquisition System* (DAQ) and a Personal Computer. The EMG signals collected at the Delsys substation are then simultaneously transferred through an unterminated DC-A22 cable to the National Instruments DAQ device. This is done so that the signals can be obtained in real time through a C++ program. The sampling rate of the DAQ and the EMG system was set at 1000 Hz. The data is pro-

cessed by the C++ program, and based on that the graphical interface is controlled.

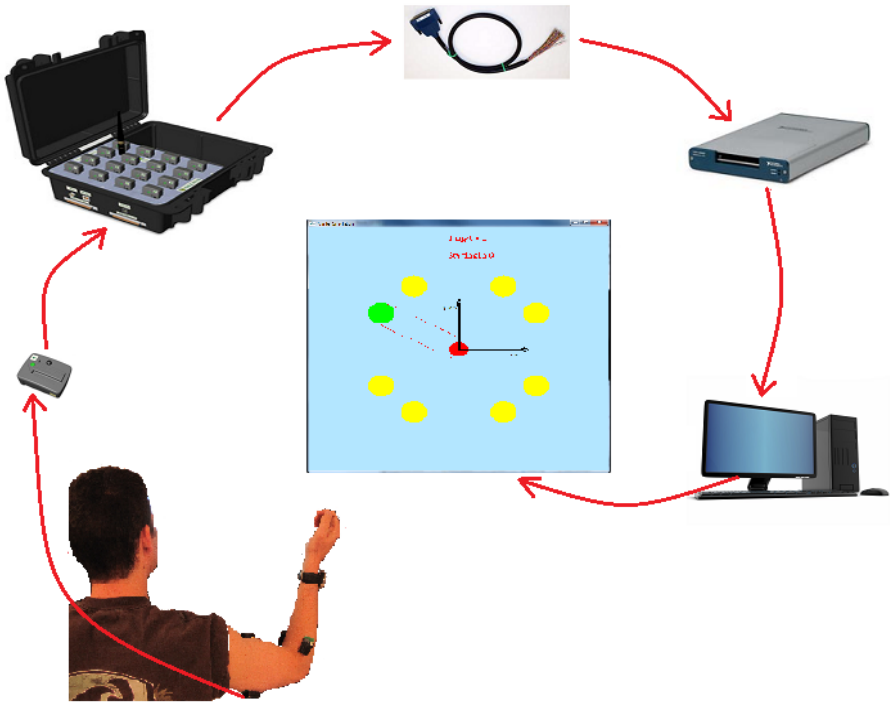

Figure 3.1: Equipment setup.

Delsys Trigno Wireless Electromyogram System consists of 16 wireless EMG sensors. Each Trigno Sensor has the following features:

1. Transmission range of 40m
2. Inter-sensor latency $< 500\mu s$ ($<$ 1 sample period)
3. EMG signal bandwidth 20- 450 Hz
4. EMG signal sampling rate up to 4000 samples/sec
5. EMG baseline noise of $< 750mV$ RMS
6. Common Mode Rejection Ratio $> 80dB$
7. 16-bit EMG signal resolution

8. Integrated triaxial accelerometer
9. Software selectable accelerometer sensitivity of $\pm 1.5g$ *or* $\pm 6g$
10. Contoured sensor-skin interface for maximum signal stability

National Instruments USB-6343 X Series Data Acquisition System consists of:

1. 32 analog inputs, 500 kS/s, 16-bit resolution, $\pm 10V$
2. Four analog outputs, 900 kS/s, 16-bit resolution, $\pm 10V$
3. 48 digital I/O lines (32 hardware-timed up to 1 MHz)
4. Four 32-bit counter/timers for PWM, encoder, frequency, event counting, and more
5. Advanced timing and triggering with NI-STC3 timing and synchronization technology

3.3 Experimental Protocol

There were four EMG electrodes placed on four specific muscles namely Biceps Brachii (BB), Triceps Brachii (TB), Flexor Carpi Ulnaris (FCU) and Extensor Carpi Radialis (ECR). Each muscle is responsible for producing unique motions of the arm. Activating the Biceps muscle produces flexion of the forearm, Triceps produces the extension of the forearm, FCU produces the flexion of the wrist and the ECR produces the extension of the wrist. The EMG sensor placement for the muscles is as shown in Fig.3.2 and is done according to literature [8].

Skin Preparation

The following procedures are steps considered to prepare the skin before electrodes are placed[22].

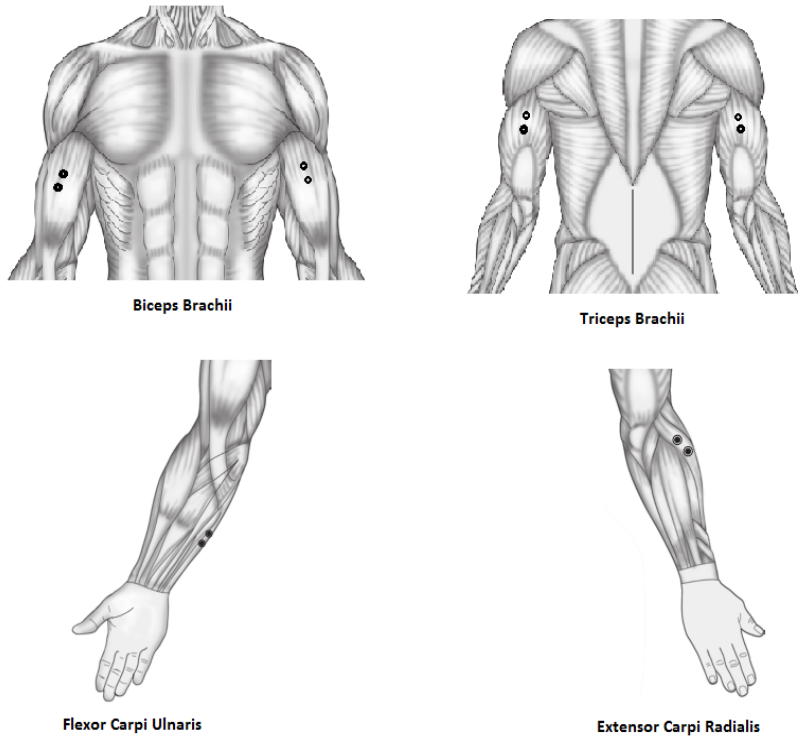

Figure 3.2: EMG Sensor location sites [8].

1. *Removing the hair:* This is needed to improve the adhesion of the electrodes, especially under humid conditions or for sweaty skin types and/or dynamic movement conditions.

2. *Cleaning of the skin:* There are three methods prescribed.

   Method A: Special abrasive and conductive cleaning pastes are available which remove dead skin cells (they produce high impedance) and clean the skin from dirt and sweat.

   Method B: Alternatively a very find sand paper can be used: A soft and controlled pressure in 3 or 4 sweeps usually is enough to get a good result. The use of sandpaper should be combined skin with an alcohol pad.

   Method C: The pure use of alcohol may be another alternative if used with a textile towel (that allows soft rubbing). This latter method may be sufficient for static muscle

function tests in easy conditions.

Behavioral Task

The experiment involved the subjects performing a center to reach out task where they had to move a cursor (round in shape and red in color) to one of the eight targets on the periphery, when instructed. The eight targets were placed at 30°, 60°, 120°, 150°, 210°, 240°, 300°, 330° with respect to the center (starting) point, as shown in Fig.3.3. Subjects were allowed to reach towards the target when one of the eight targets changed their color from yellow to green and simultaneously a beep sound is generated, signaling start of the trial. Subjects were also instructed to move along the direction of target as close as possible. The workspace consists of an $x-y$ plane which ranges from -50 to +50 graphical units on both axis. A successful attempt would be when the subject is within a square area of side 4 units centered around the middle of the target, and stays within that region for 500 milliseconds. Targets were all presented on a 2 dimensional plane.

In terms of controlling the cursor motion, subjects had no information on how to move the cursor in the two dimensional space. The only information given was how to activate the muscles by doing specific motions of the arm. The experiment was designed in C++ , and the real time graphics were programmed by using OpenGL libraries. EMG signals were collected in real time from the C++ code using the National Instruments Daqmx library. The x - y positions and the time stamps for each trial are saved to a text file for analysis.

Subjects

Experiments were performed on 2 human subjects. There were two sessions, each consisting of approximately one hour, and with a gap of 2 days between the sessions. All subjects gave informed consent which was according to the procedures approved by the ASU IRB (Protocol: 1201007252).

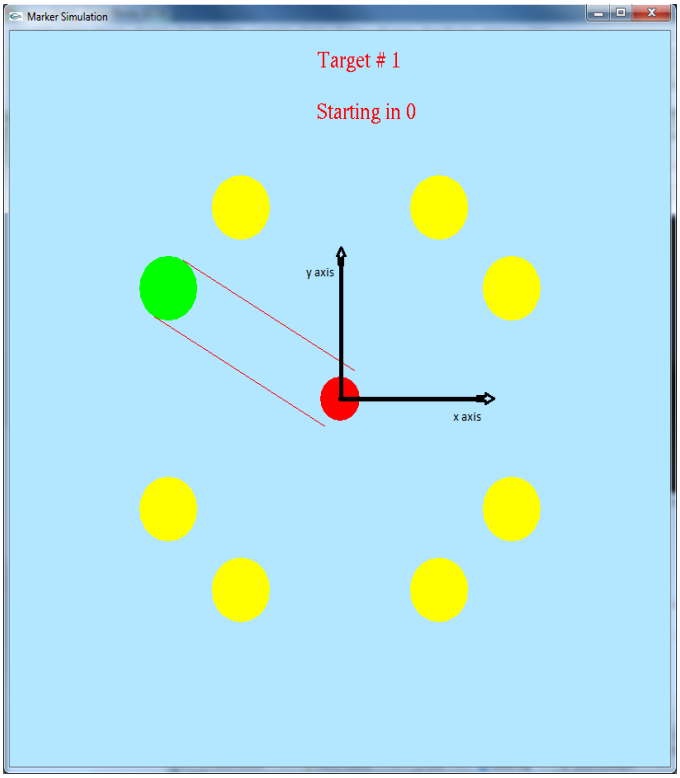

Figure 3.3: Graphical interface of the experiment.

Experiments

There were two kinds of experiments namely A and B.

Experiment A: It was done in three sets, each set had 8 trials for each of the eight targets, resulting in a total of 24 trials per target. Sequence of trials to the eight targets were generated in a random fashion, such that two successive trials to the same target happens only after one complete cycle of 8 different targets. The cursor was controlled by activating specific combinations of the four muscles. Each muscle controlled motion in a specific direction which is as shown in Table.3.1.

Motion in a particular direction was activated once the muscle activity crossed a particular threshold level. The threshold to all the four muscle activities were fixed at .04 millivolts. When the subject didn't do any motion, or when they were relaxing, the EMG

Table 3.1: Muscles and their control axes-directions

Muscle	Axis of Motion	Direction (+ or -)
Biceps brachii (BB)	x axis	-
Triceps Brachii (TB)	x axis	+
Flexor Carpi Ulnaris (FCU)	y axis	-
Extensor Carpi Radialis (ECR)	y axis	+

activation was less than the threshold levels and the cursor would stay fixed. If for instance the biceps muscle was activated and once the signal value crosses the threshold, the cursor would move along the x axis, in the negative direction. Processed EMG signals were not normalized using the maximum voluntary contraction (MVC) level of the respective muscles. The motion commands from each muscle are updated every 1 millisecond. The cursor would therefore keep on moving towards the left as long as it exceeds its threshold level. Simultaneous contraction of two or more muscles above their threshold limits was similar to performing vectors addition to determine the cursor position. Care was taken to ensure that the cursor always stayed within the workspace window.

$$X_t = \begin{cases} X_{t-1} - \Delta x, & u_t^1 > 0.04 \\ X_{t-1}, & u_t^1 \leq 0.04 \end{cases} \tag{3.1}$$

$$X_t = \begin{cases} X_{t-1} + \Delta x, & u_t^2 > 0.04 \\ X_{t-1}, & u_t^2 \leq 0.04 \end{cases} \tag{3.2}$$

$$Y_t = \begin{cases} Y_{t-1} - \Delta y, & u_t^3 > 0.04 \\ Y_{t-1}, & u_t^3 \leq 0.04 \end{cases} \tag{3.3}$$

$$Y_t = \begin{cases} Y_{t-1} + \Delta y, & u_t^4 > 0.04 \\ Y_{t-1}, & u_t^4 \leq 0.04 \end{cases} \tag{3.4}$$

where X_t and Y_t are the x and y coordinates of the cursor at time step t, X_{t-1} and Y_{t-1} are the x and y coordinates of the cursor at time step $t-1$, $u_t^1, u_t^2, u_t^3, u_t^4$ are the amplitude of the processed EMG signal for BB, TB, FCU and ECR, at time step t, Δx and Δy are the position increments in the x and y axis respectively.

Experiment B: This experiment, instead of using individual muscles mapped to individual axial motion, it uses the low-dimensional representation of muscle activation to control the cursor. In order to do that, we use the PCA method, as described in section2.4. An intermediate experiment was done between experiment A and B, where the subjects were asked to perform natural motion tasks involving all joints of the arm. This was done so as to find out the synergistic patterns in the activation of the four muscles. The data is processed using the PCA method and the explained variance is calculated for each of the eigenvectors. All the subjects had a cumulative explained variance of more than 85% with just two eigenvectors. The plot in Fig.3.4 shows the cumulative explained variance plot for subject 1. Hence, the new dimensional axis representation was reduced to just two components, i.e. the value for n as shown in Deriving the New Data Set section5, was chosen to be equal to 2. Once these steps are done, the *feature vector* and the mean EMG value of each muscle are noted. These values are used in the C++ code for real time processing. The output F is a 2×1 vector (reduced dimensional data) at each time step. The output vector is multiplied by gain factor to adjust the speed of the cursor motion. A higher level controller is implemented to ensure that motion is produced only when EMG signal values are above the threshold level set in the previous experiment. This experiment was also done in three sets, each set had 8 trials corresponding to each of the eight targets, resulting in a total of 24 trials per target. We wanted to see if a reduced dimensionality would still be able to produce the same extent of learning and retainment of the subject control skill.

$$F_t = \begin{cases} V^T H_t, & u_t^1, u_t^2, u_t^3, u_t^4 > 0.04 \\ [0\,0\,0\,0]^T, & u_t^1, u_t^2, u_t^3, u_t^4 \leq 0.04 \end{cases} \tag{3.5}$$

where V is the 4×2 feature vector, $u_t^1, u_t^2, u_t^3, u_t^4$ are the amplitude of the processed EMG signal (without normalization using the MVC of each muscle) for BB, TB, FCU and ECR, at time step t, H_t is the 4×1 vector containing the mean subtracted EMG values of the four muscles at time t, F_t is the reduced dimensional output vector.

$$X_t = \{X_{t-1} + K_1 F_t^1\} \tag{3.6}$$

$$Y_t = \{Y_{t-1} + K_2 F_t^2\} \tag{3.7}$$

where X_t and Y_t are the x and y coordinates of the cursor at time step t, K_1 and K_2 are the gain values for adjusting the speed of the cursor in the x and y directions respectively, F_t^1 and F_t^2 are the first and second component in the reduced dimensional output vector.

It must be noted that both subjects completed first experiment A, and after two days, they were used for experiment B. For experiment B, they were not told whether they would do the same experiment or not.

3.4 Data Pre-processing

All EMG signals were processed in real time before they could be decoded to cursor position. EMG signals contain a lot of high frequency sensor noise, which should be removed. The main steps involved in the processing of EMG signals are:

1. *Full wave rectification of EMG signal:*

   The first step is to take the absolute value of the EMG signal. It is represented by

$$|x| = \begin{cases} x, & x \geq 0 \\ -x, & x < 0 \end{cases} \tag{3.8}$$

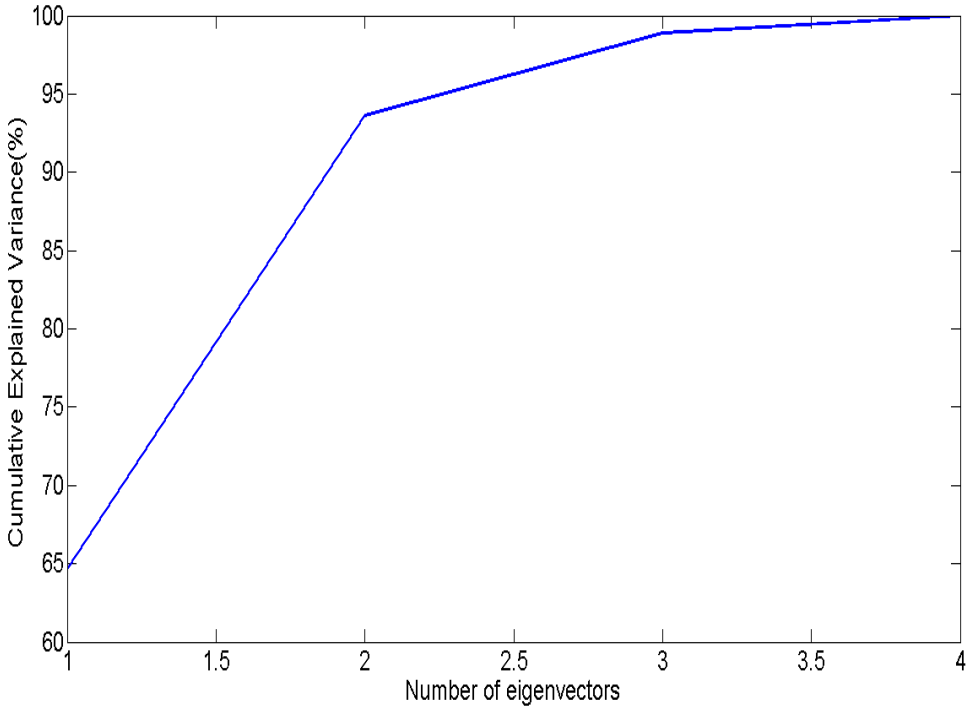

Figure 3.4: Cumulative explained variance using the 4 eigenvectors.

2. *Lowpass Filtering:*

   The second step is to obtain a linear envelope of the rectified EMG signal to get information about the amplitude. The amplitude gives an idea of how hard the subject is flexing his/her muscles. For that, EMG signal is passed through a low pass filter to filter out the high frequency noise. A second order butterworth filter is designed in MATLAB. It is then implemented in C++ code by using the relation,

$$uf_i^n = \frac{b_3 u_i^{n-2} + b_2 u_i^{n-1} + b_1 u_i^n - a3 uf_i^{n-2} - a_2 uf_i^{n-1}}{a_1} \tag{3.9}$$

   where uf represents the filtered signal, u represents the unfiltered signal, i represents the i^{th} EMG signal, n represents the n^th time step, b_1, b_2, b_3 represent the coefficients of the numerator in the filter transfer function, and a_1, a_2, a_3 represent the coefficients of the denominator in the filter transfer function.

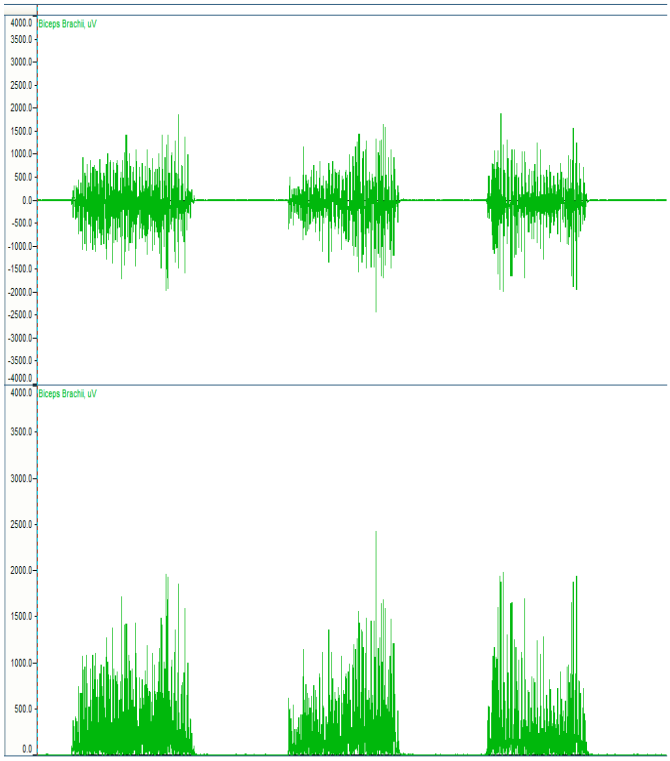

Figure 3.5: EMG rectification[22]

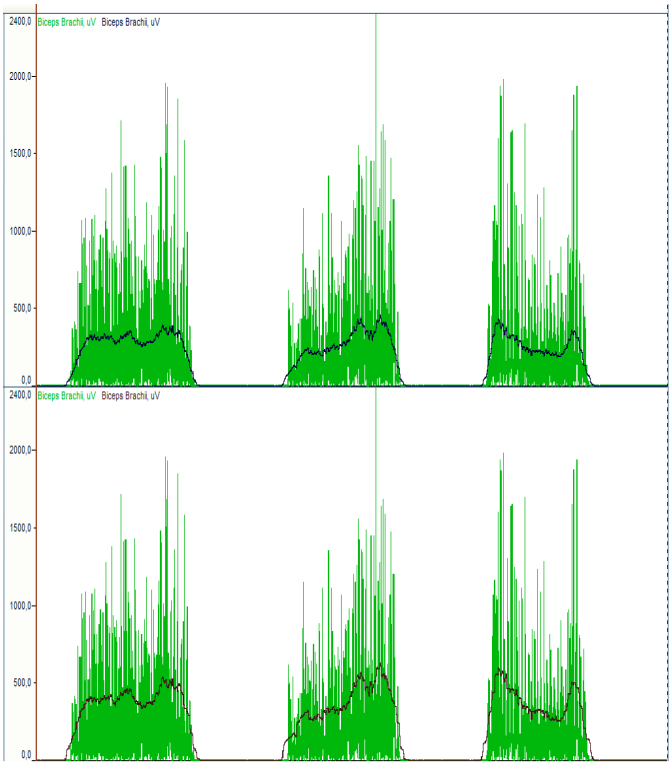

Figure 3.6: Linear Envelope of rectified EMG [22].

Chapter 4

Data Analysis and Results

4.1 Introduction

There were basically two kinds of experiments (A and B), which were mainly aimed at how well subjects could learn the inverse model of an abstract mapping function. As such, we need to see whether they were able to successfully complete each task that was given to them, and if they do, how fast were they able to learn the task. For that we need to analyze the learning curve associated with each target. Another aspect of analyzing the data, is to see the smoothness of the motion, that is how accurate was the subject in moving towards the target. Also how far did the subject deviate from the desired trajectory in order to reach the target, needs to be considered. Then looking into the EMG signals we need to see how well they are coordinated as the subject learns to control the motion tasks. This would be an implication of how muscles develop synergies while learning to perform a task. The second set of experiments (experiment B) used a reduced dimensionality mapping to control the reaching motion. Therefore, we need to see how subjects performed in that case and also make a comparison between experiment A and experiment B.

4.2 Criteria for Assessment of Motion Control

The different criteria based on which learning the inverse model of an abstract mapping function was determined is a follows:

1. *Learning Curve:* The learning curve is a plot showing the time taken to reach the same target for the various trials. A successful effort by the subject in learning the particular task is quantified if the time taken to reach a particular target gradually reduces. Also in order to determine if the skill that is being mastered is retained and not temporary, there should a steady state in the learning curve as the number of trials towards the same target increases, as shown in Fig.4.1.

2. *Smoothness of Motion:* Smoothness is another factor which needs to be quantified.

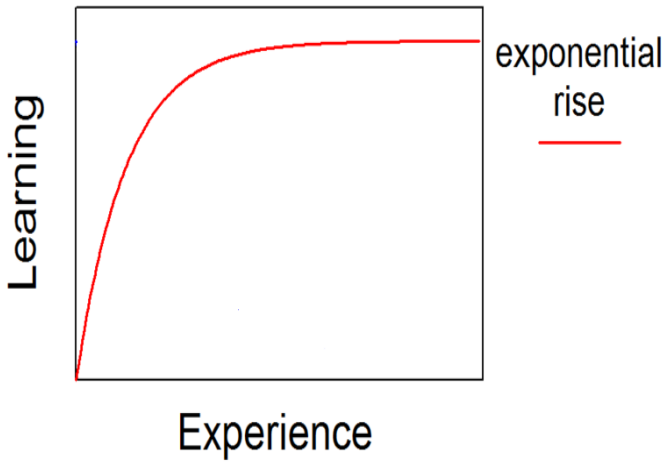

Figure 4.1: Learning Curve

The experiment is designed such that activation of one muscle alone is not necessary to reach the target. Activation of an individual muscle would only result in motion along the x or y axis of the workspace. Targets were well placed inside the quadrants such that the subjects need to activate more than one muscle simultaneously to reach the target. Activating individual muscles in succession would also result in reaching towards the target, but in that case the path followed would not be in the direction of target. The direction towards each target or the slope is considered the baseline, and is compared to the average slope that the subject traverses to reach each specific target. Fig.4.2 shows the preferred direction in red line in order to reach the target. Targets are arranged at 30°,60°,120°,150°,210°,240°,300°,330°.

3. *Deviation from the Preferred direction:* This factor is an extension of the previous one. Here we quantify how far the subject deviated from the preferred direction of motion. This is done by taking the standard deviation of the motion from the preferred slope, while reaching towards a particular target. During the course of learning when the subject is able to produce a smooth motion, the standard deviation should be gradually reducing. The standard deviation is given by:

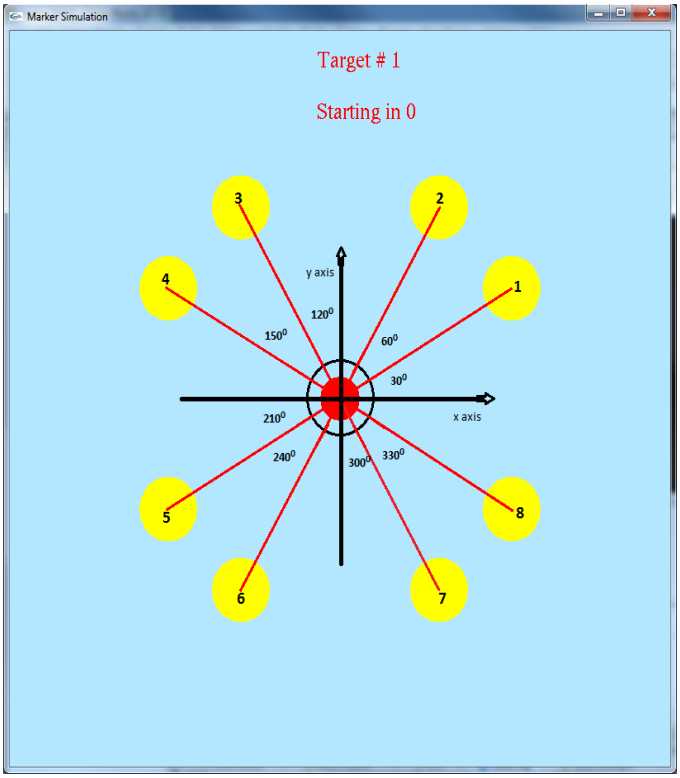

Figure 4.2: Preferred direction of motion

$$\sigma = \sqrt{\frac{\sum_{i=1}^{N}(X_i - \bar{X})^2}{N}} \tag{4.1}$$

where X_i is the slope at time step i , N is the total number of time steps for that particular motion task, $\bar{X}$ is the preferred slope for that particular target.

4. *Development of Synergies:* This criterion is based on the EMG signal of the muscles. Learning any task involves synergistic combination of muscles. The way to identify this is by finding out the explained variance of the eigenvectors in the covariance matrix between the muscles. If we see that there is an increasing trend in the explained variance from just one or two principal eigenvectors, then we can say that the subject is developing muscle synergies while learning the task. The covariance matrix is given by:

$$cov(X,Y) = \frac{\sum_{i=1}^{N}(X_i - \bar{X})(Y_i - \bar{Y})}{N} \tag{4.2}$$

where X and Y are two dimensions across which variance is to be calculated, $\bar{X}$ and $\bar{Y}$ are the mean values of X and Y respectively.

4.3 Results

Experiment A

For each of the targets 1 to 8, four plots were generated namely A) Time taken in each trial to reach the target (Top left), B) Average slope during each trial in comparison with the Preferred slope (Top right), C) Standard deviation from the Preferred Slope in each trial (Bottom left), and D) Explained Variance of the principal Eigenvectors (Bottom right). The plots are shown below: Fig.5.1, Fig.5.2, Fig.5.3, Fig5.4, Fig.5.5, Fig5.6, Fig.5.7, Fig.5.8. These are from subject 1.

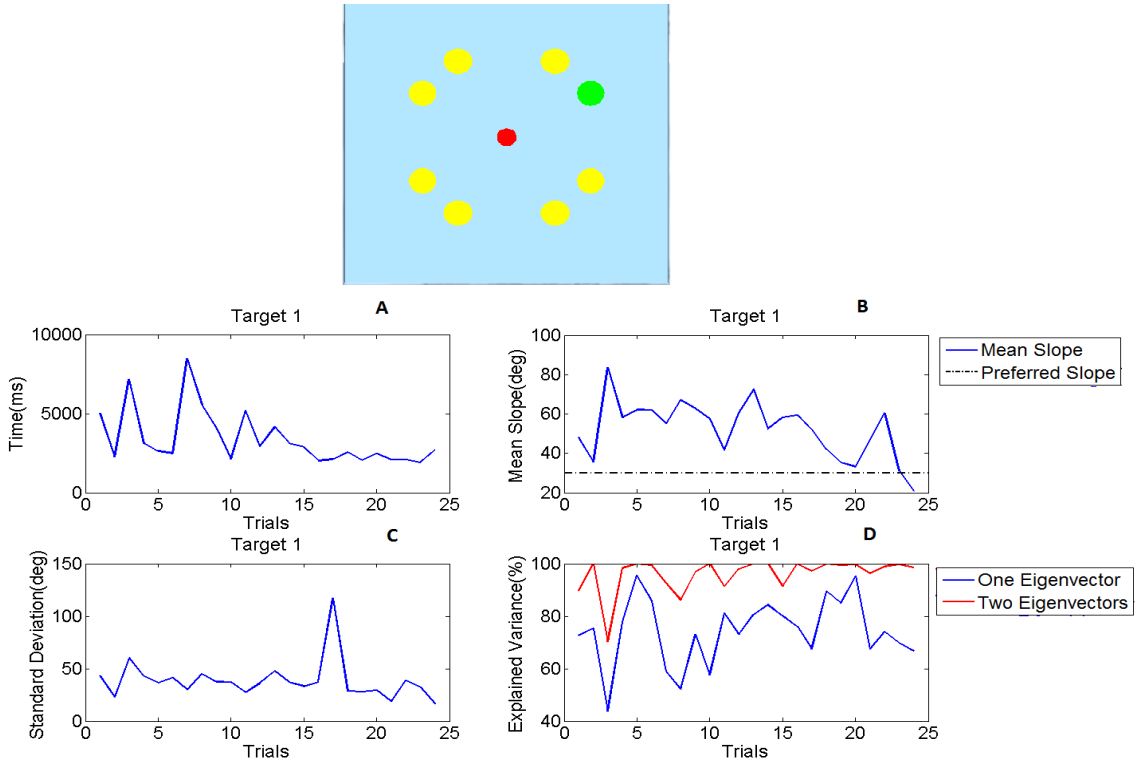

Figure 4.3: A: Time plot, B: Average slope plot, C: Standard deviation plot, D: Explained variance plot, for Target 1 (Experiment A)

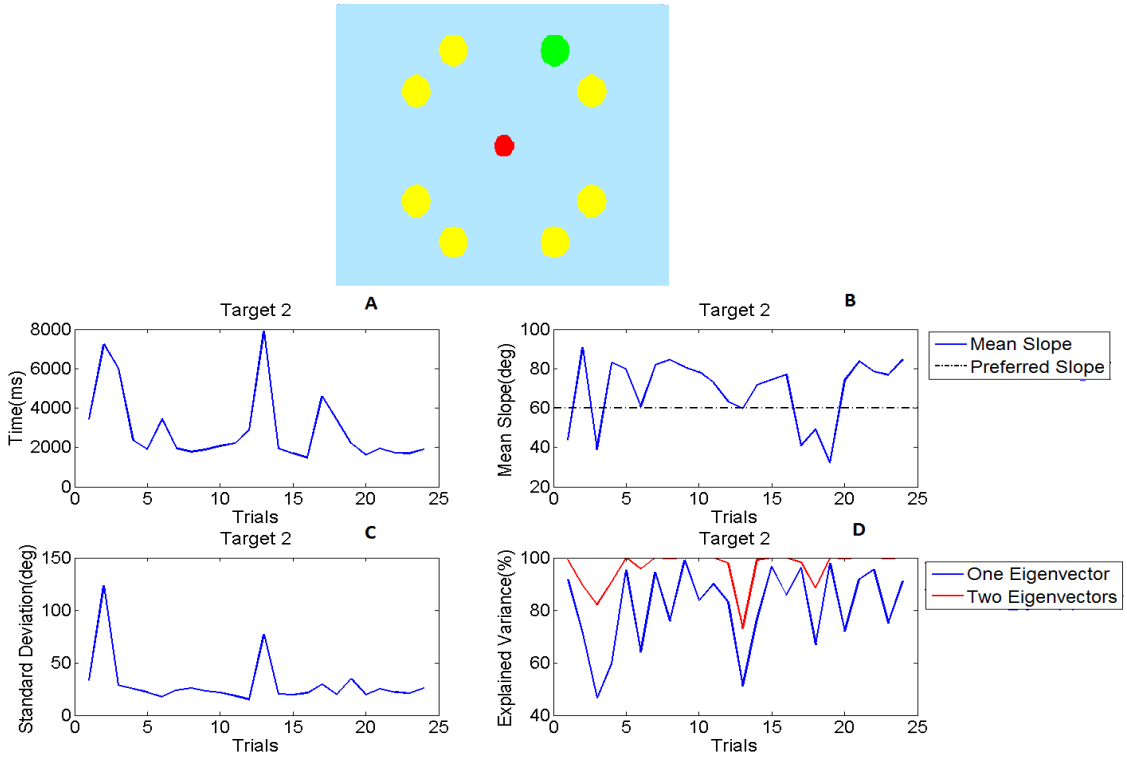

Figure 4.4: A: Time plot, B: Average slope plot, C: Standard deviation plot, D: Explained variance plot, for Target 2 (Experiment A)

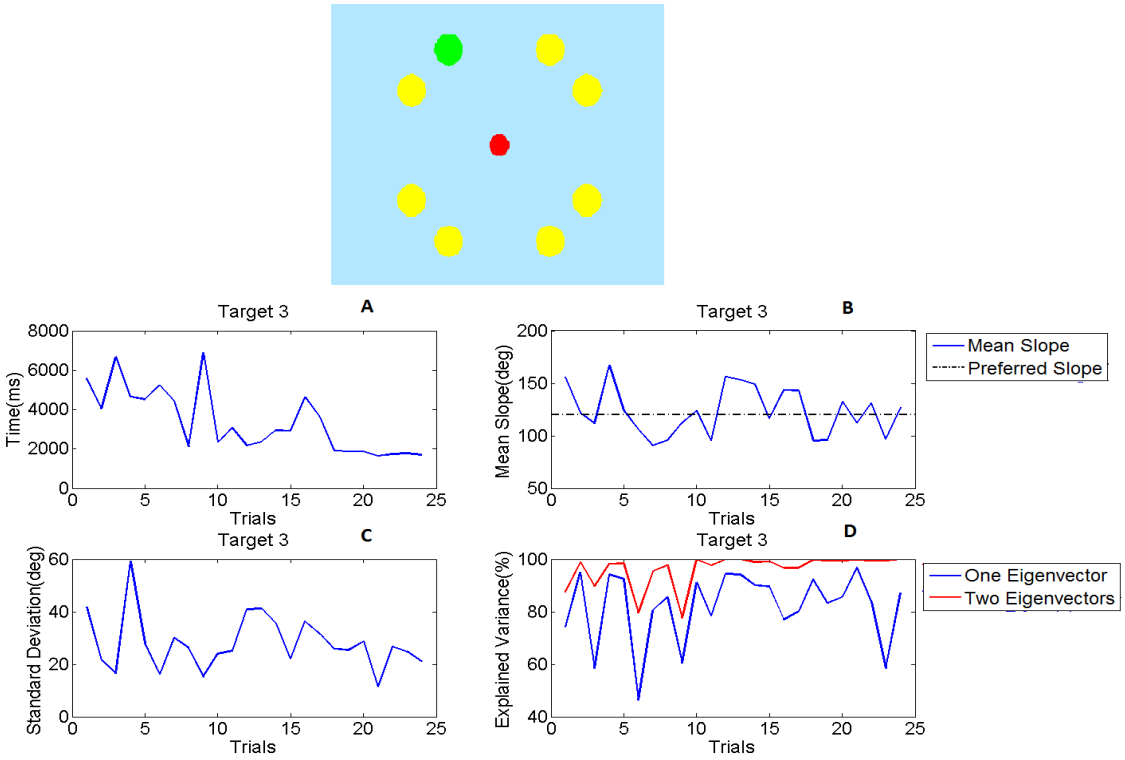

Figure 4.5: A: Time plot, B: Average slope plot, C: Standard deviation plot, D: Explained variance plot, for Target 3 (Experiment A)

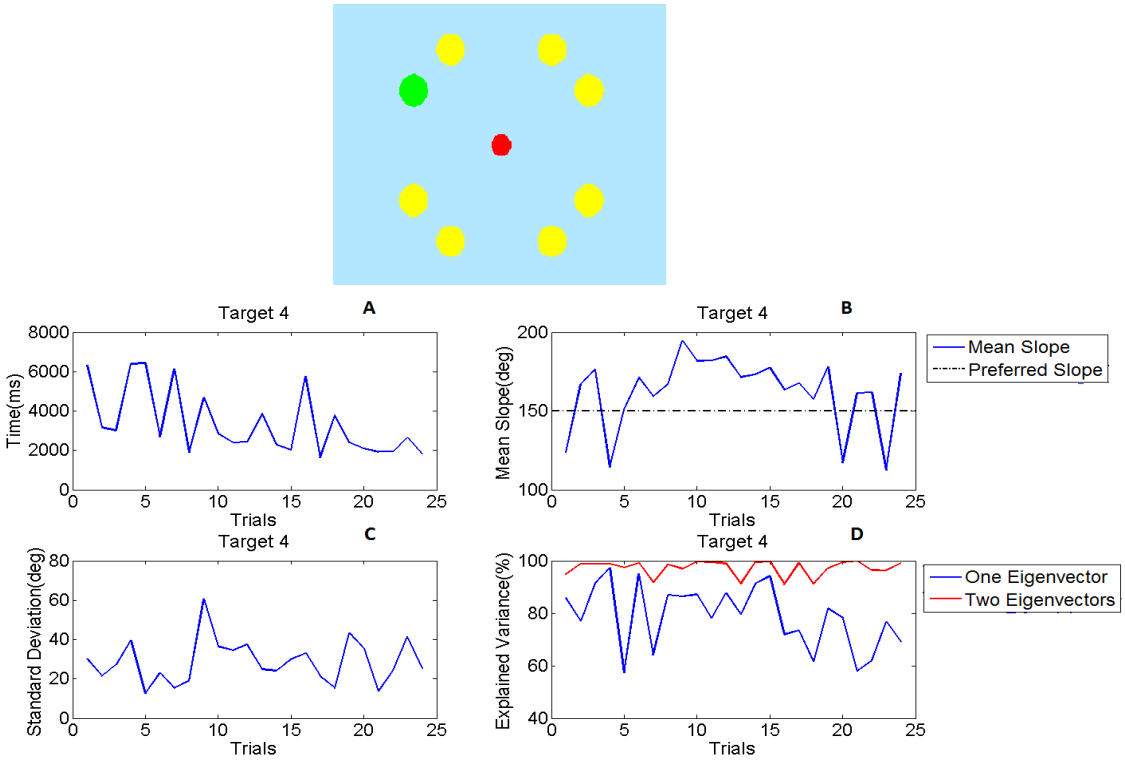

Figure 4.6: A: Time plot, B: Average slope plot, C: Standard deviation plot, D: Explained variance plot, for Target 4 (Experiment A)

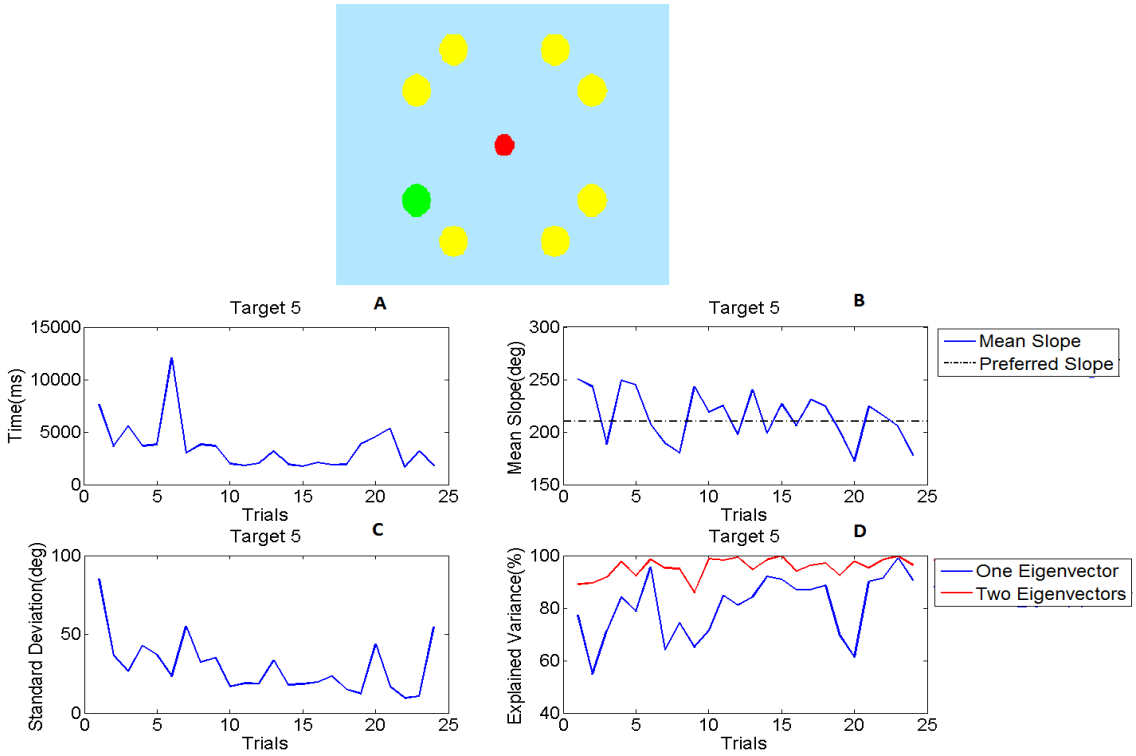

Figure 4.7: A: Time plot, B: Average slope plot, C: Standard deviation plot, D: Explained variance plot, for Target 5 (Experiment A)

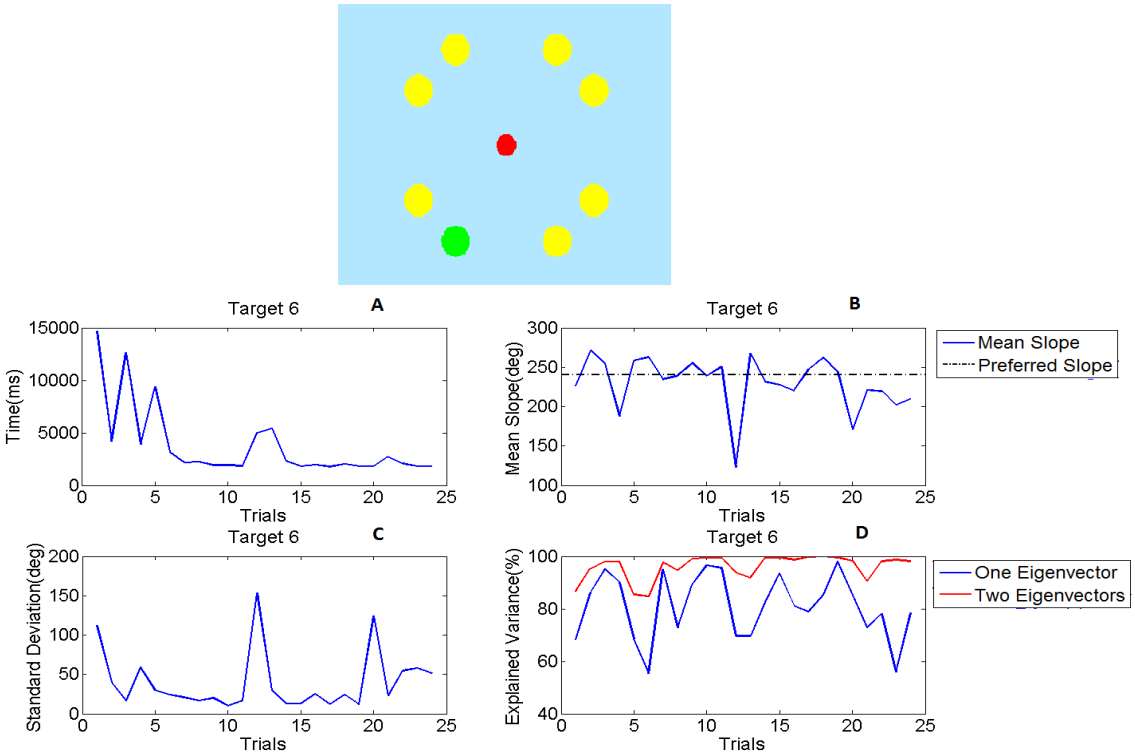

Figure 4.8: A: Time plot, B: Average slope plot, C: Standard deviation plot, D: Explained variance plot, for Target 6 (Experiment A)

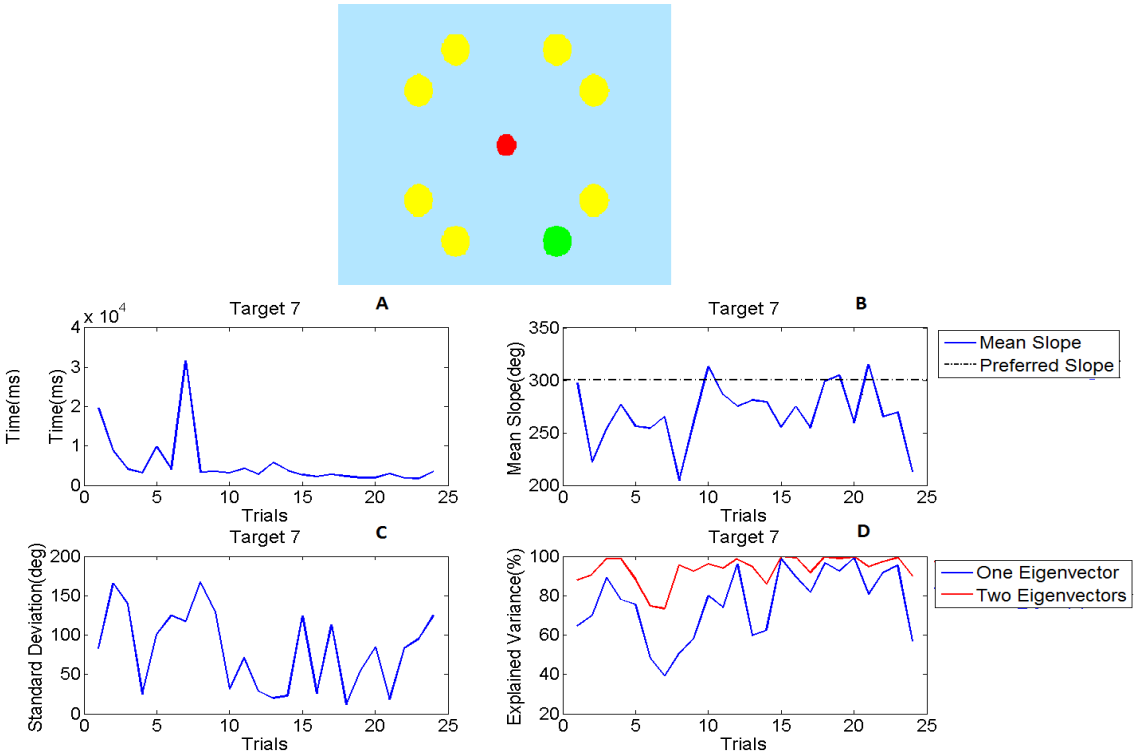

Figure 4.9: A: Time plot, B: Average slope plot, C: Standard deviation plot, D: Explained variance plot, for Target 7 (Experiment A)

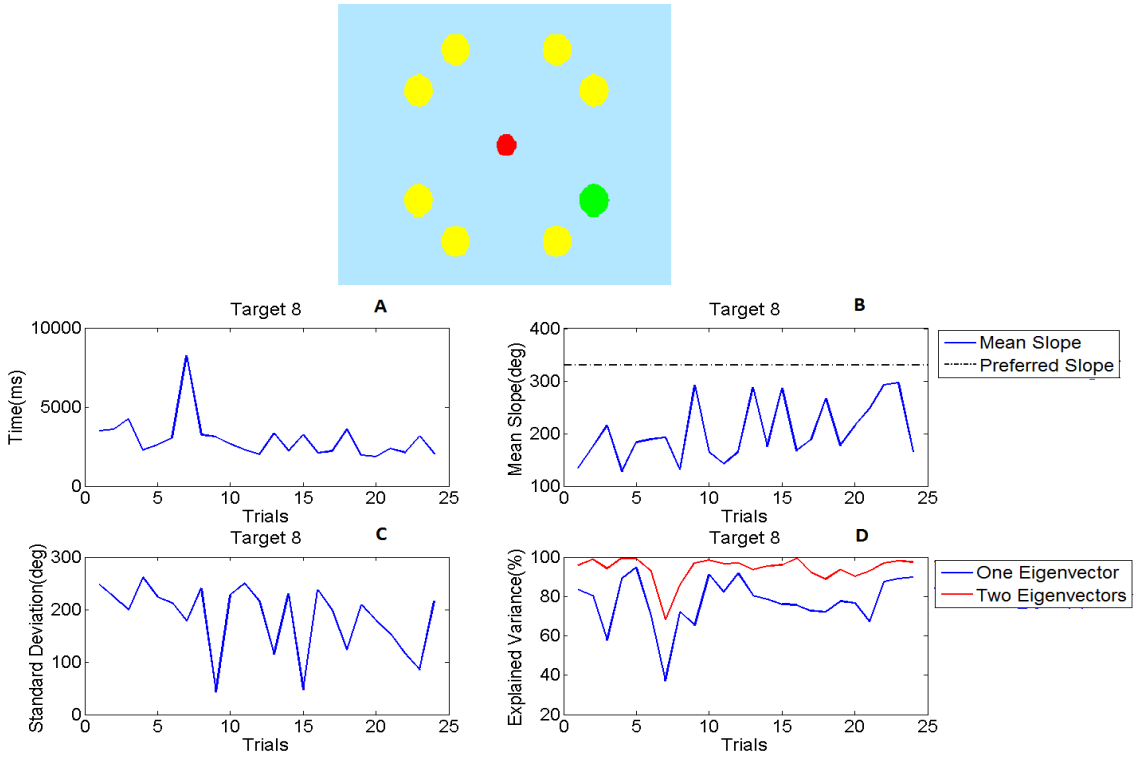

Figure 4.10: A: Time plot, B: Average slope plot, C: Standard deviation plot, D: Explained variance plot, for Target 8 (Experiment A)

1. *Time Analysis:* The time plot which is the subplot A in each of the eight targets, shows a gradual decrease in the time taken to reach the target. This shows that the subject is slowly learning the task, and he/she is able to proceed towards the target faster with each passing trial. Also another interesting thing to notice is the attainment of steady state indicating that the subject has learned the task. The time taken to reach the target once they attain the steady state, i.e. when the subject learns to control the task better, is less than 2 seconds, for almost all the targets.

2. *Smoothness Analysis:* As can be seen in the subplot B, the mean slope, in most of the cases, was not exactly matching the preferred slope, although it converged very close to the preferred slope. It shows that the subject is trying to move more streamlined towards the target. It is fair to say that, the more the subject learns to move in a streamlined path, the better controllability skills he/she has developed. The plots do show a decrease in the error between the mean slope and the preferred slope, but there is tolerance level within which the subject seems to have developed controllability. Also if we check the standard deviation plot C, we notice that the deviation from the preferred direction also shows signs of improvement.

3. *Synergistic developments:* If we check subplot D, which shows the explained variance of the eigenvectors, we notice that there is not a specific trend. More specifically, there are no increasing trends in the explained variance, which shows that the subject is not developing synergies. This trend was observed in both the subjects used. The reason for this is that this experiment is based on individual muscle control. Therefore the subjects do not develop synergistic muscle activation while learning to perform the task in this experiment.

Experiment B

Similarly to the data analysis of experiment A, there are four plots generated for each of the eight targets, which includes A) Time taken in each trial to reach the target (Top left),

B) Average slope during each trial in comparison with the preferred slope (Top right), C) Standard deviation from the preferred slope in each trial (Bottom left) and D) Explained variance of the eigenvectors (Bottom right). Moreover, the corresponding plots from experiment A were superimposed to make a comparison between the two experiments. The plots are shown in Fig.5.9, Fig.5.10, Fig.5.11, Fig.5.12, Fig.5.13, Fig.5.14, Fig5.15, Fig.5.16. The data from experiment A is shown in blue color, and the data from experiment B is shown in red color.

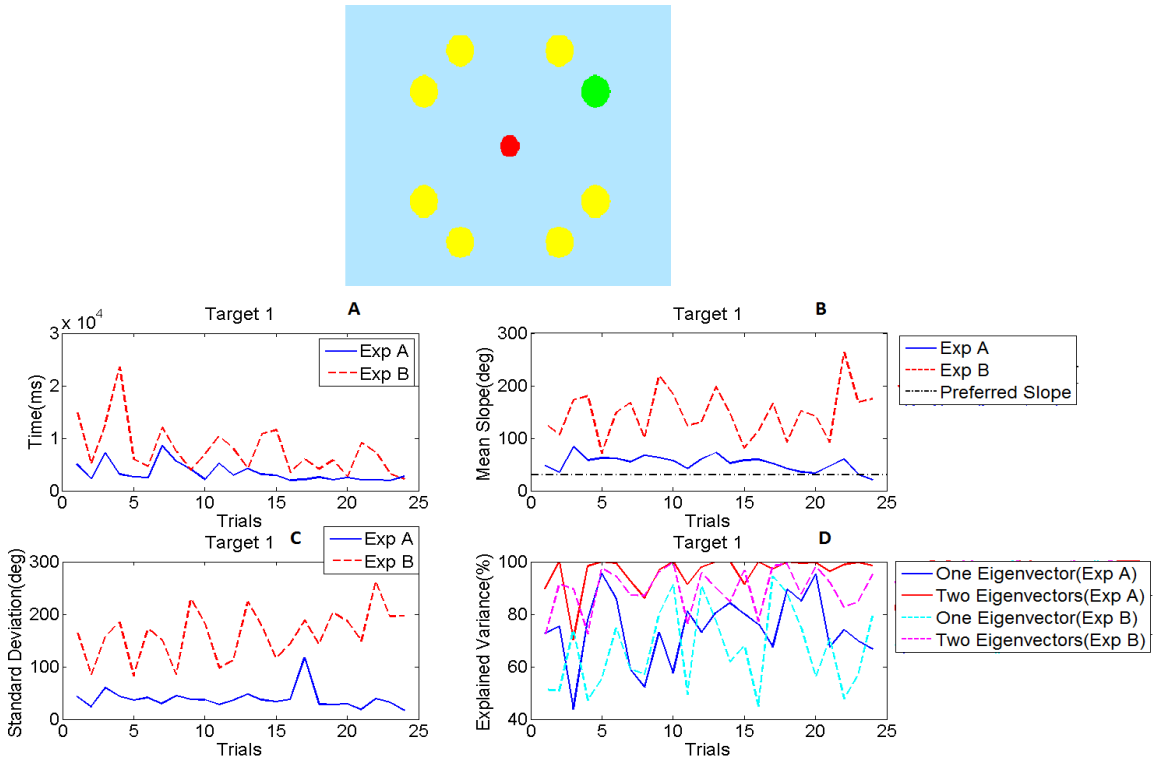

Figure 4.11: A: Time plot, B: Average slope plot, C: Standard deviation plot, D: Explained variance plot, for Target 1 (Experiment A and B)

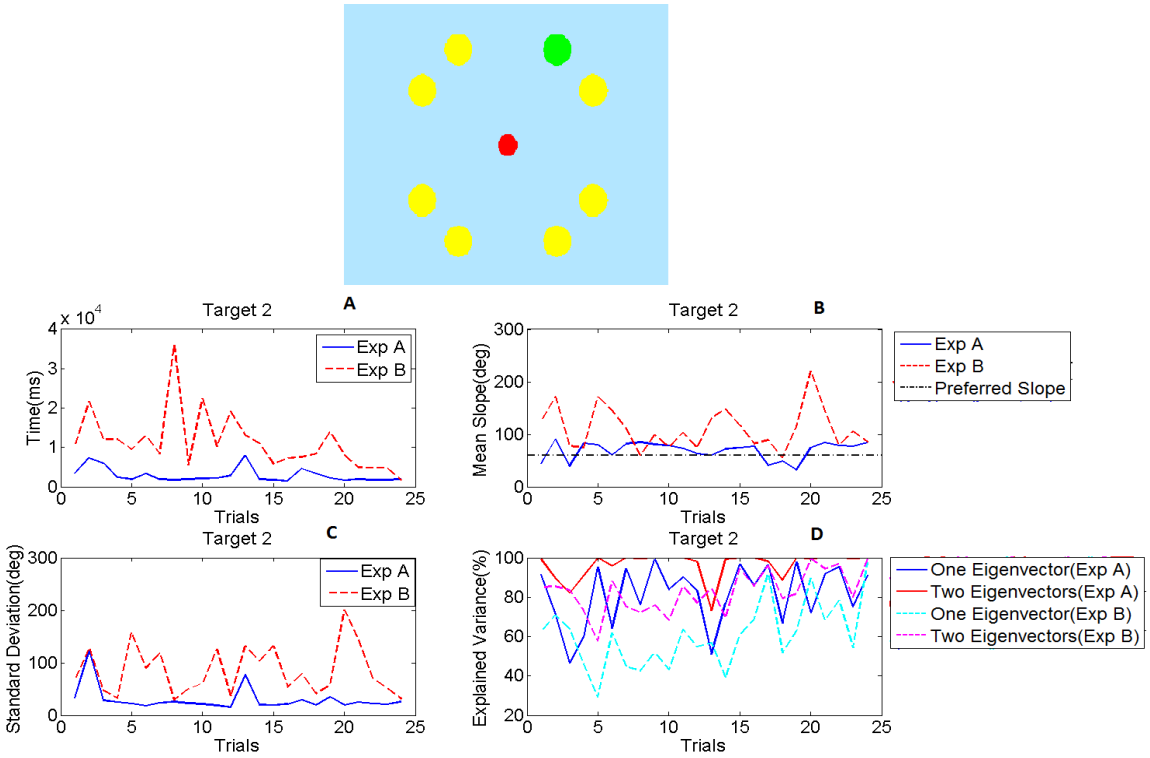

Figure 4.12: A: Time plot, B: Average slope plot, C: Standard deviation plot, D: Explained variance plot, for Target 2 (Experiment A and B)

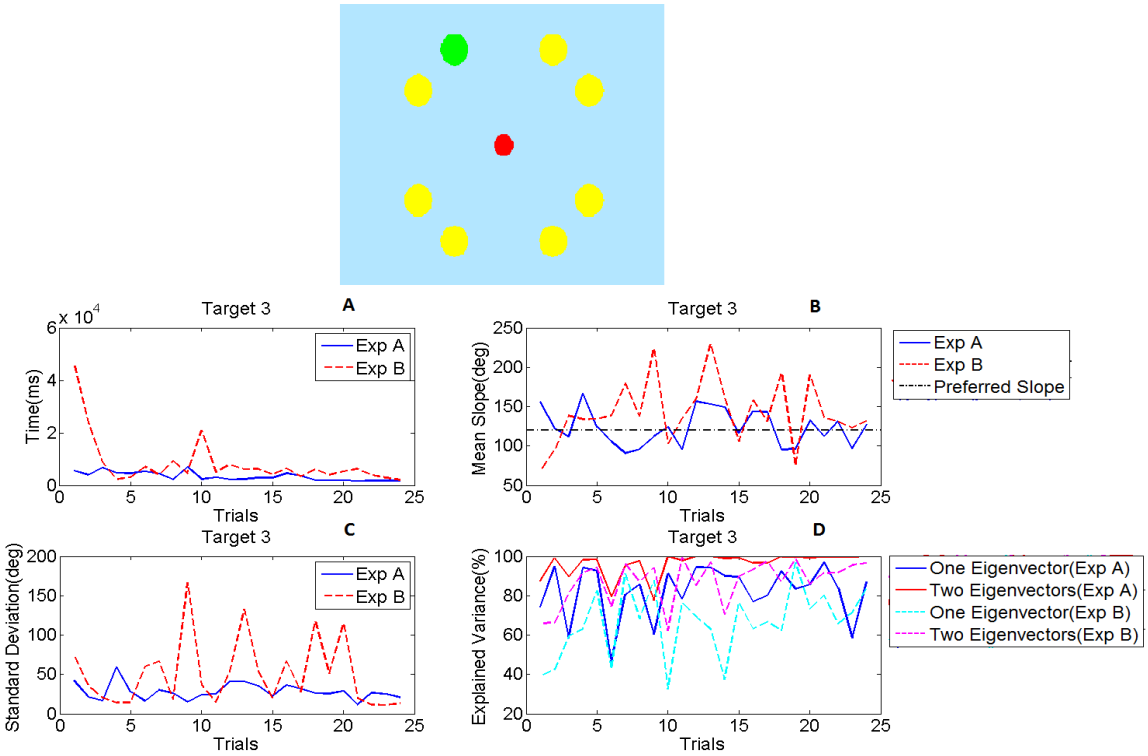

Figure 4.13: A: Time plot, B: Average slope plot, C: Standard deviation plot, D: Explained variance plot, for Target 3 (Experiment A and B)

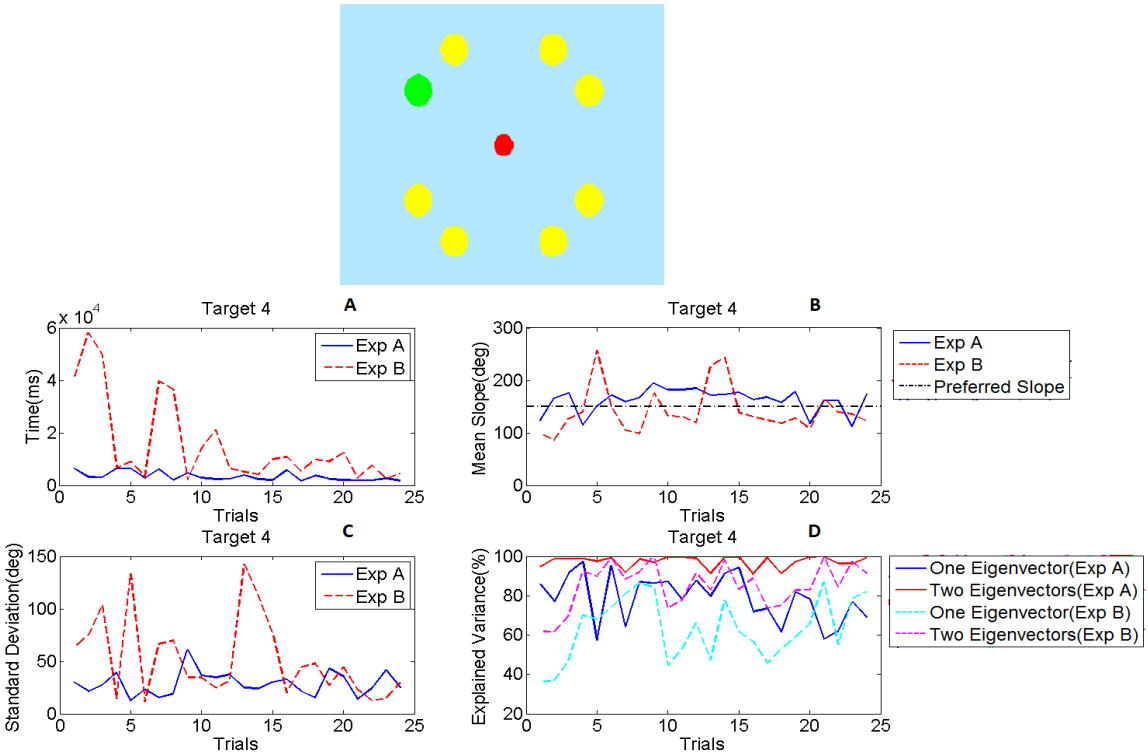

Figure 4.14: A: Time plot, B: Average slope plot, C: Standard deviation plot, D: Explained variance plot, for Target 4 (Experiment A and B)

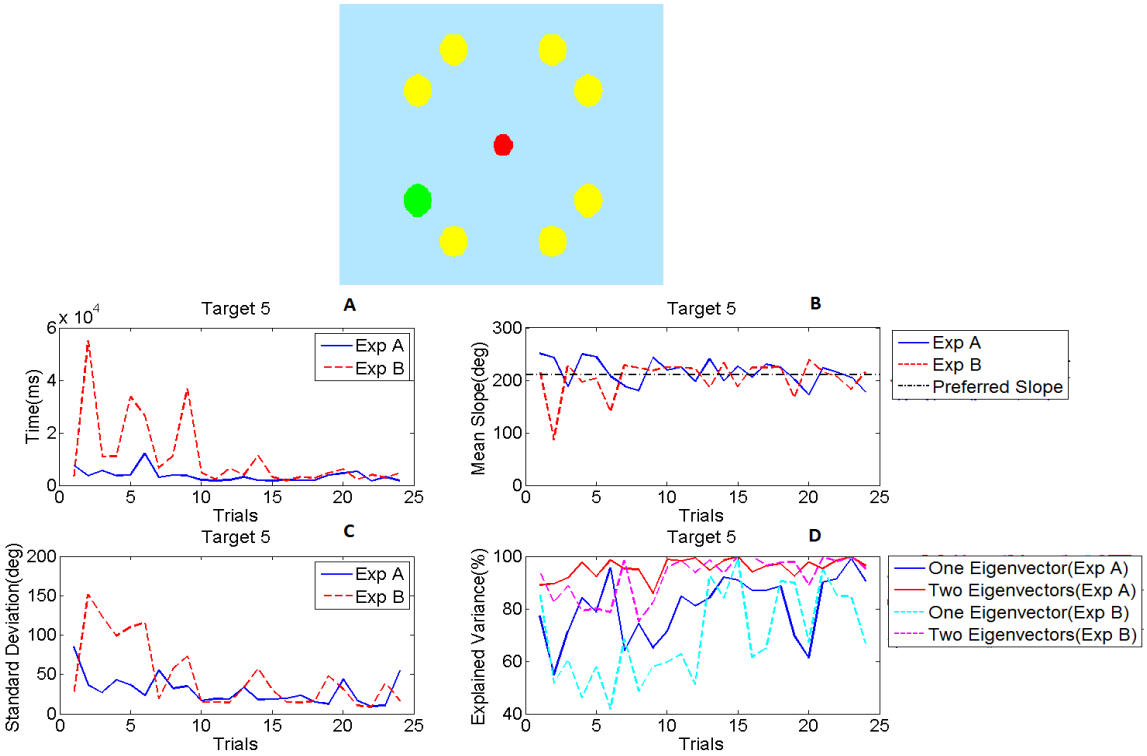

Figure 4.15: A: Time plot, B: Average slope plot, C: Standard deviation plot, D: Explained variance plot, for Target 5 (Experiment A and B)

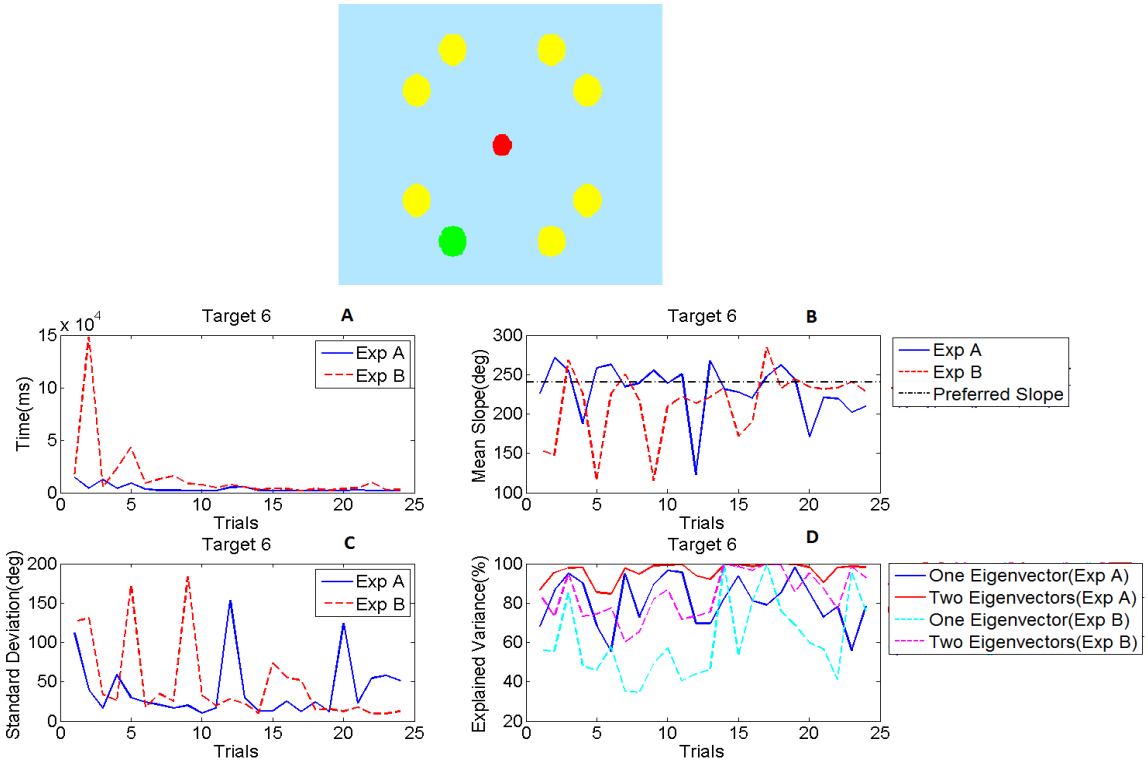

Figure 4.16: A: Time plot, B: Average slope plot, C: Standard deviation plot, D: Explained variance plot, for Target 6 (Experiment A and B)

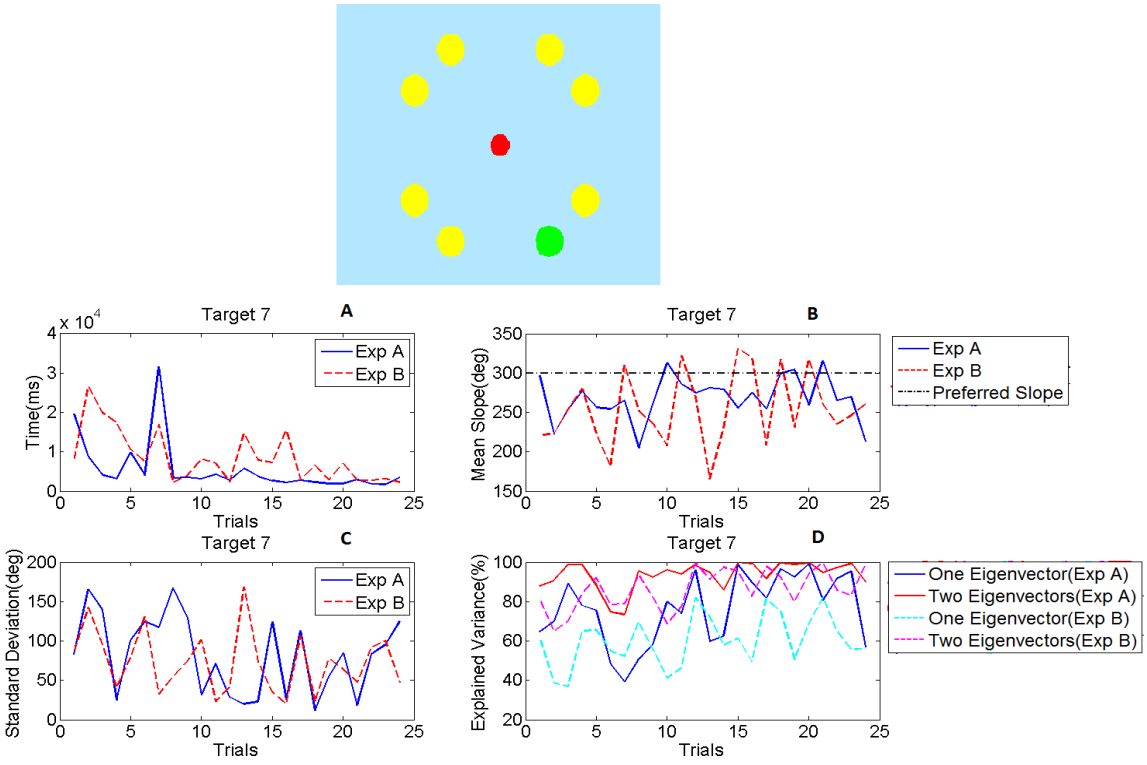

Figure 4.17: A: Time plot, B: Average slope plot, C: Standard deviation plot, D: Explained variance plot, for Target 7 (Experiment A and B)

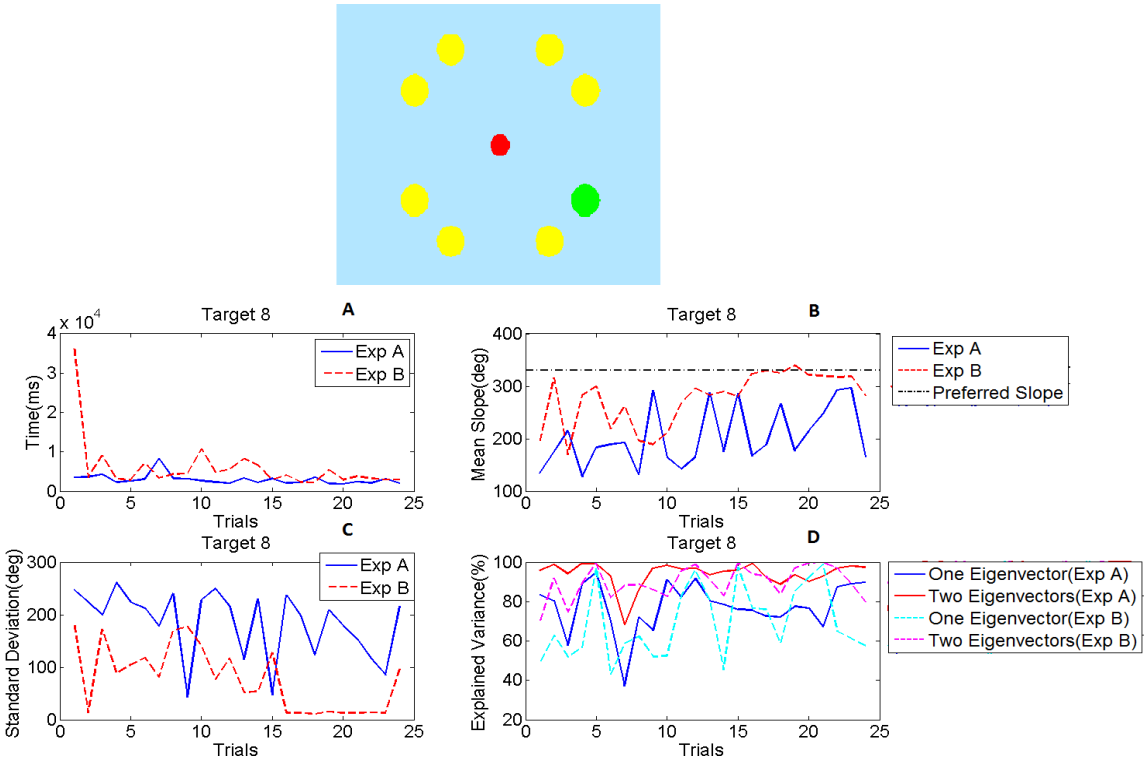

Figure 4.18: A: Time plot, B: Average slope plot, C: Standard deviation plot, D: Explained variance plot, for Target 8 (Experiment A and B)

1. *Time Analysis:* If we analyze the plots, we notice that the subjects take a lot of time during the initial stages in experiment B to reach the target, when compared to those trials in experiment A. But as the number of trials towards the same target increases, the time taken to complete the task is almost the same in both the experiments. The learning curve is a bit less steep in this case but finally converges around the same steady state value. This shows that subjects are also able to attain and retain the same level of learning in the low dimensional representation case (Experiment B).

2. *Smoothness Analysis:* In the subplot B for the figures in experiment B, we notice that the mean slope to a target moves closer to the preferred slope as the number of trials increases. When compared to experiment A, the average slope and standard deviation plots of experiment B are slightly poor. But it is encouraging to see slightly better performance towards the end. These plots indicate that subjects are also able to control the motion of the cursor using the synergistic, low dimensional representation of the muscles.

3. *Synergistic developments:* In the case of experiment B we notice that the explained variance for one and two eigenvectors shows a much better increasing trend when compared to experiment A, which indicates the development of synergies in muscle activations. This is a general trend that was noticed across all targets.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

As indicated in the objective, we wanted to analyze learning in two different kinds of experiments. In Experiment A, motion tasks were based on individual muscle control. Subjects were able to learn the abstract task and learning saturated at some point in time indicating that the subject has learned to control the motion task. But synergistic developments were not so obvious from the plots of explained variance.

In the next set of trials, Experiment B, we conclude that even though the initial learning period is high, the subjects finally attain the same level of learning as in Experiment A. Although the average slope and standard deviation plots show poor performance in experiment B when compared to experiment A, there seems to be small improvement towards the end of the 24 trials. We need to increase the number of trials and do more experiments to arrive at more better conclusions. We also notice some improvement in explained variance indicating development of synergies. Developing synergistic muscle patterns are essential in optimizing the control structure for EMG based robot control.

5.2 Future Directions

The next step towards making EMG-based robot control interfaces is bringing the control interface closer to reality.

1. We definitely need to do more experiments and increase the number of trials to see more pronounced difference between the two experiments. Also we need to do investigate if subjects are able to retain the mapping they have learned across days.

2. The first step in the project would be to implement the experiment in the three dimensional space and identify if subjects are capable of controlling objects in that space.

If the subjects learn to be dexterous in maneuvering objects in the Cartesian space, then it will be a great leap in myoelectric based control.

3. Another future objective would be to incorporate the subject's impedance level to the object that is being controlled. This is done by measuring the EMG signal levels. Impedance control is very important for a successful human machine interface to be implemented and its addition to the EMG-based control interfaces is of paramount importance.

REFERENCES

[1] Panagiotis K. Artemiadis and Kostas J. Kyriakopoulos. Emg-based teleoperation of a robot arm in planar catching movements using armax model and trajectory monitoring techniques. *IEEE International Conference on Robotics and Automation*, 1:6, 2006. vi, 11, 12

[2] Panagiotis K. Artemiadis and Kostas J. Kyriakopoulos. Emg-based teleoperation of a robot arm using low-dimensional representation. *IEEE International Conference on Intelligent Robots and Systems*, 1:7, 2007. 21

[3] Sebastian Bitzer and Patrick van der Smagt. Learning emg control of a robotic hand: Towards active prostheses. *IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION*, 1:5, 2006. 1

[4] B. Boser, I. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, 1:9, 1992. 9

[5] Jose M. Carmena, Mikhail A. Lebedev, Roy E. Crist, Joseph E. ODoherty, David M. Santucci, Dragan F. Dimitrov, Parag G. Patil, Craig S. Henriquez, and Miguel A. L. Nicolelis. Learning to control a brainmachine interface for reaching and grasping by primates. *PLOS BIOLOGY*, 1:16, 2003. 15

[6] Steven M. Chase, Andrew B. Schwartz, and Robert E. Kass. Bias, optimal linear estimation, and the differences between open-loop simulation and closed-loop performance of spiking-based brain computer interface algorithms. *Neural Networks*, 22:11, 2009. 15

[7] Philip Chrapka. Emg controlled hand prosthesis: Emg classification system. *McMaster University*, 1:40, 2010. vi, 10, 11

[8] Eleanor Criswell. Cram's introduction to surface electromyography. 1:436, 2011. vi, 26, 27

[9] John P. Cunningham, Paul Nuyujukian, Vikash Gilja, Cindy A. Chestek, Stephen I. Ryu, and Krishna V. Shenoy. A closed-loop human simulator for investigating the role of feedback control in brain machine interfaces. *J Neurophysiol*, 105:17, 2011. vi, 2, 16, 17

[10] Andrea dAvella, Philippe Saltiel, and Emilio Bizzi1. Combinations of muscle synergies in the construction of a natural motor behavior. *Nature Neuroscience*, 1:9, 2003. vi, 20, 21

[11] Osamu Fukuda, Toshio Tsuji, Makoto Kaneko, and Akira Otsuka. A human-assisting manipulator teleoperated by emg signals and arm motions. *IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION*, 19:13, 2003. 1

[12] Osamu Fukuda, Toshio Tsuji, Akira Ohtsuka, and Makoto Kaneko. Emg-based human-robot interface for rehabilitation aid. *IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION*, 1:6, 1998. 1

[13] Santhanam G, Ryu SI, Yu BM, Afshar A, and Shenoy KV. A high-performance brain computer interface. *Nature*, 442:4, 2006. 15

[14] Gregory J Gage, Kip A Ludwig, Kevin J Otto, Edward L Ionides, and Daryl R Kipke. Naive coadaptive cortical control. *JOURNAL OF NEURAL ENGINEERING*, 1:13, 2005. vi, 2, 13, 15

[15] Karunesh Ganguly and Jose M. Carmena. Emergence of a stable cortical map for neuroprosthetic control. *PLOS BIOLOGY*, 1:13, 2009. 15

[16] D. Graupe and W. K. Cline. Functional separation of emg signals via arma identification methods for prosthesis control purposes. *Prentice Hall*, 1:842, 1999. 5

[17] Simon S. Haykin. Neural networks: A comprehensive foundation. *Prentice Hall*, 1:842, 1999. 9

[18] Rodolphe Hliot, Karunesh Ganguly, Jessica Jimenez, and Jose M. Carmena. Learning in closed-loop brainmachine interfaces: Modeling and experimental validation. *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS*, 40:11, 2010. vi, 12, 13, 14

[19] Yonghong Huang, Kevin B. Englehart, Bernard Hudgins, and Adrian D. C. Chan. A gaussian mixture model based classification scheme for myoelectric control of powered upper limb prostheses. *IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING*, 52:11, 2005. 7

[20] B. Hudgins, P. Parker, and R. Scott. A new strategy for multifunction myoelectric control. *IEEE Transactions on Biomedical Engineering*, 40:13, 1993. vi, 5, 9, 10

[21] J. Edward Jackson. A users guide to principal components. *John Wiley & Sons*, 1:569, 1991. 21

[22] Peter Konrad. The abc of emg:a practical introduction to kinesiological electromyography. 1:60, 2005. vi, 5, 6, 26, 34, 35

[23] Carlo J. De Luca. The use of surface electromyography in biomechanics. *The International Society for Biomechanics*, 1:38, 1993. vi, 6, 8

[24] Saritha M Radhakrishnan, Stuart N Baker, and Andrew Jackson. Learning a novel myoelectric-controlled interface task. *J Neurophysiol*, 1:47, 2008. vi, 16, 18, 19

[25] Kim S, Simeral JD, Hochberg LR, Donoghue JP, and Black MJ. Neural control of computer cursor velocity by decoding motor cortical spiking activity in humans with tetraplegia. *J Neural*, 5:22, 2008. 15

[26] Koyama S, Chase SM, Whitford AS, Velliste M, Schwartz AB, and Kass RE. Comparison of brain-computer interface decoding algorithms in open-loop and closed-loop control. *J Computational Neuroscience*, 29:15, 2010. 15, 16

[27] Daniel L. Schacter, Daniel T. Gilbert, and Daniel M. Wegner. Psychology 2nd edition. 1:264, 2011. 3

[28] Dawn M. Taylor, Stephen I. Helms Tillery, and Andrew B. Schwartz. Direct cortical control of 3d neuroprosthetic devices. *Science*, 296:5, 2002. vi, 2, 14, 16

[29] Jonathan R. Wolpaw, Niels Birbaumer, Dennis J. McFarland, Gert Pfurtscheller, and Theresa M. Vaughan. Braincomputer interfaces for communication and control. *Clinical Neurophysiology*, 113:25, 2002. 1

[30] Jonathan R. Wolpaw, Dennis J. McFarland, Gregory W. Neat, and Catherine A. Forneris. An eeg-based brain-computer interface for cursor control. *Clinical Neurophysiology*, 78:8, 1991. 1

[31] Felix E. Zajac. Muscle and tendon: properties, models, scaling, and application to biomechanics and motor control. *Stanford University*, 17:52, 1989. 2

APPENDIX

Plots from Subject 2

Experiment A

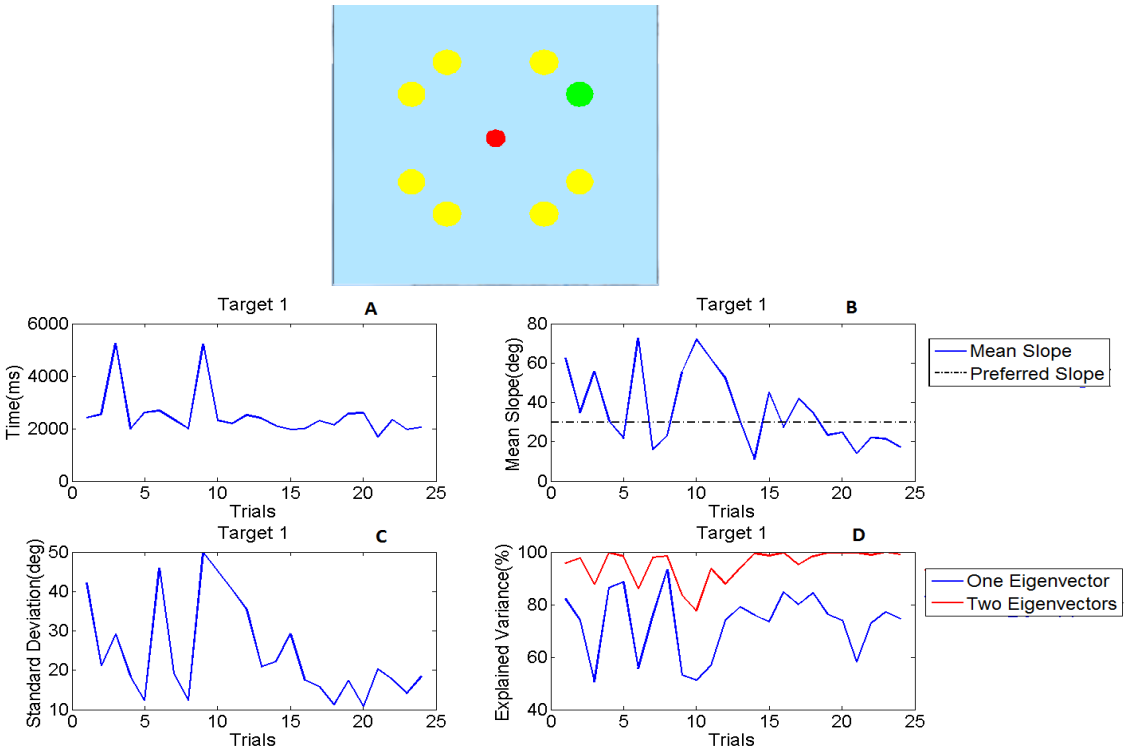

Figure 5.1: A: Time plot, B: Average slope plot, C: Standard deviation plot, D: Explained variance plot, for Target 1 (Experiment A)

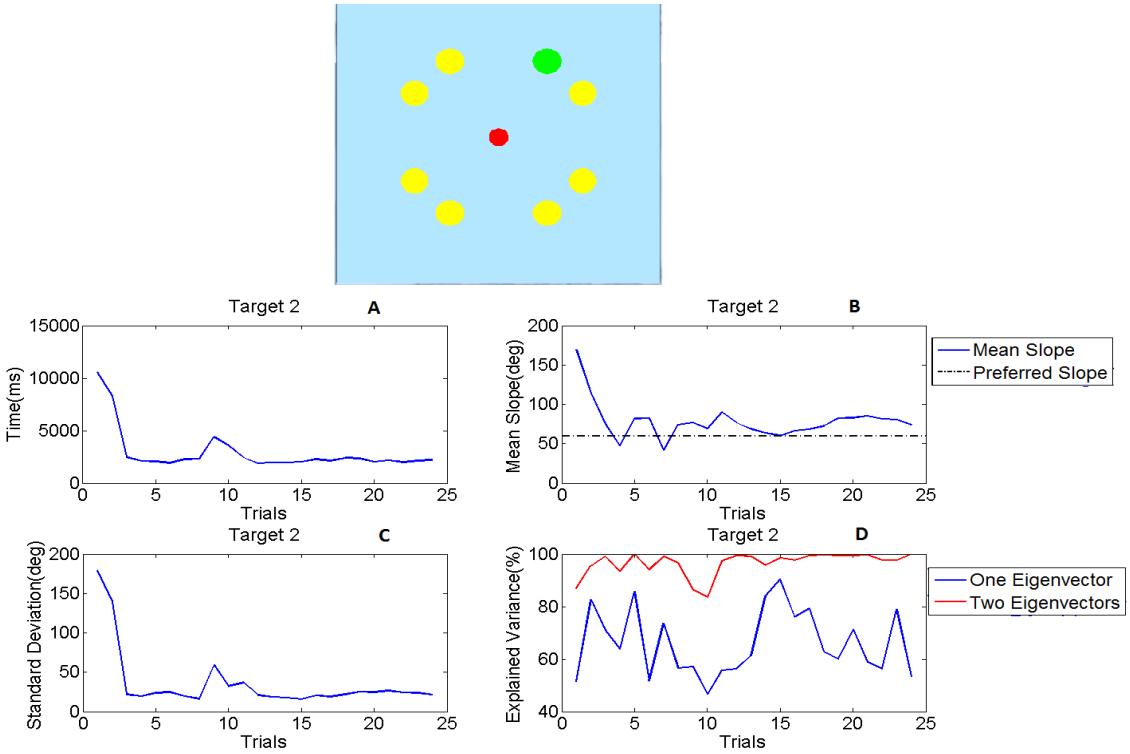

Figure 5.2: A: Time plot, B: Average slope plot, C: Standard deviation plot, D: Explained variance plot, for Target 2 (Experiment A)

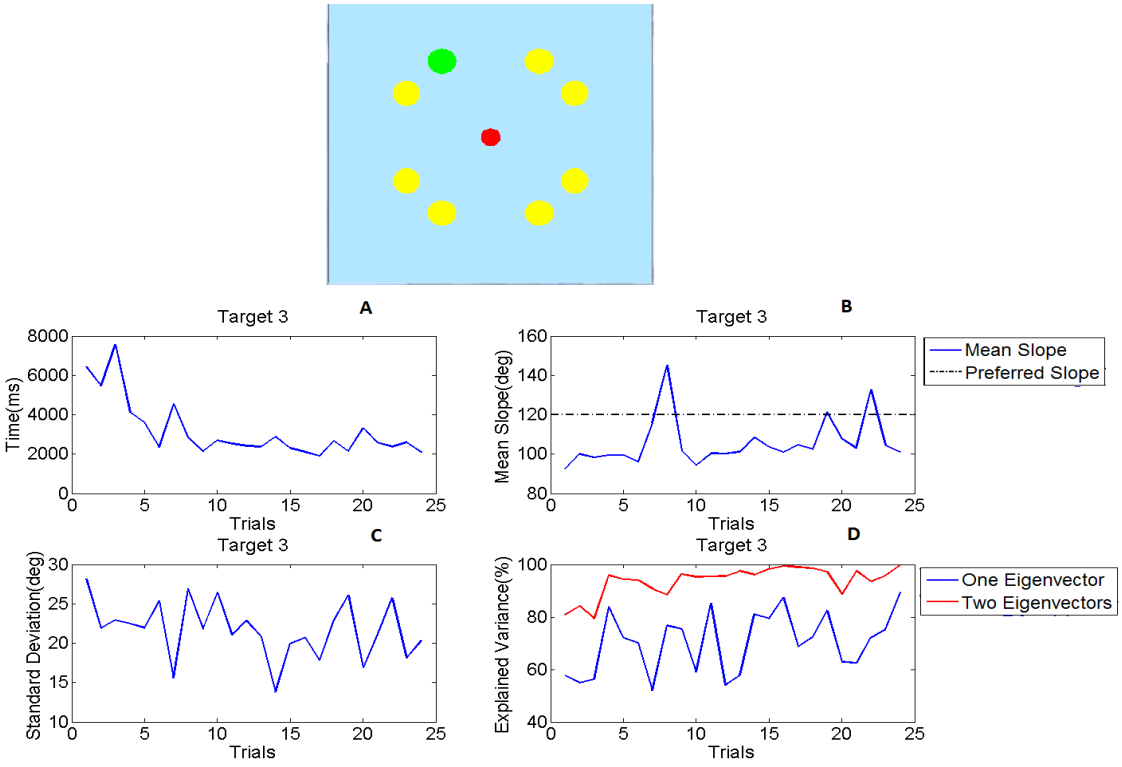

Figure 5.3: A: Time plot, B: Average slope plot, C: Standard deviation plot, D: Explained variance plot, for Target 3 (Experiment A)

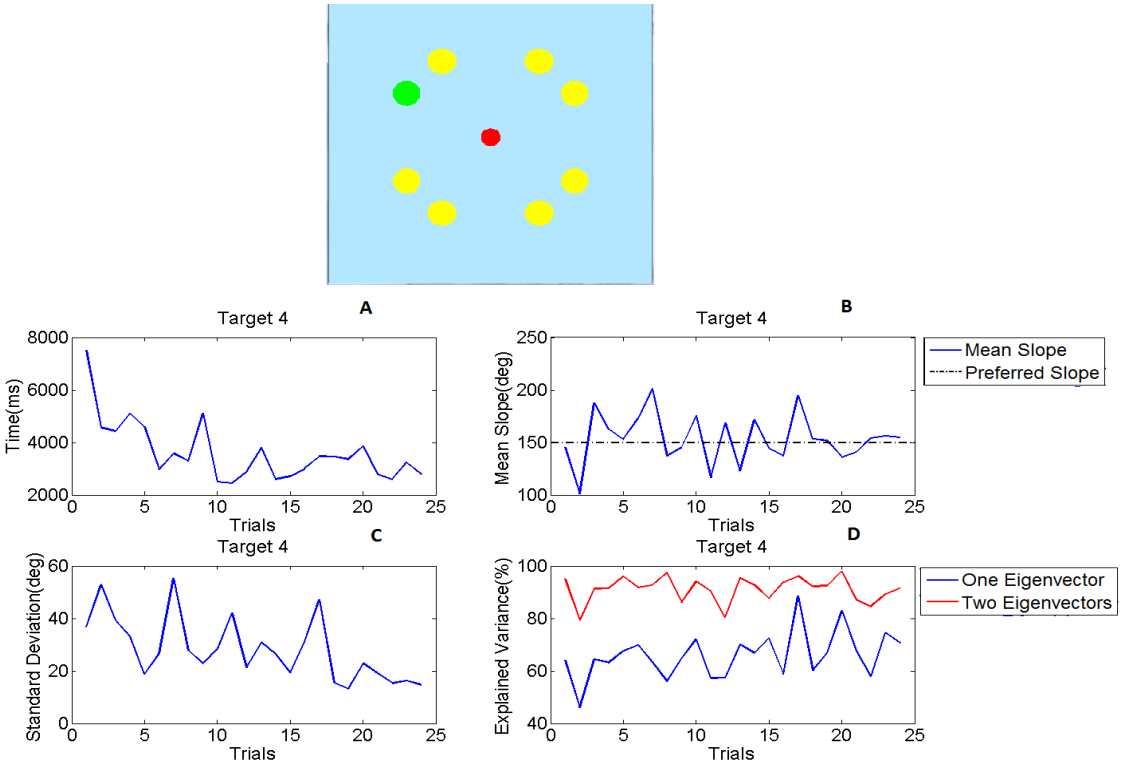

Figure 5.4: A: Time plot, B: Average slope plot, C: Standard deviation plot, D: Explained variance plot, for Target 4 (Experiment A)

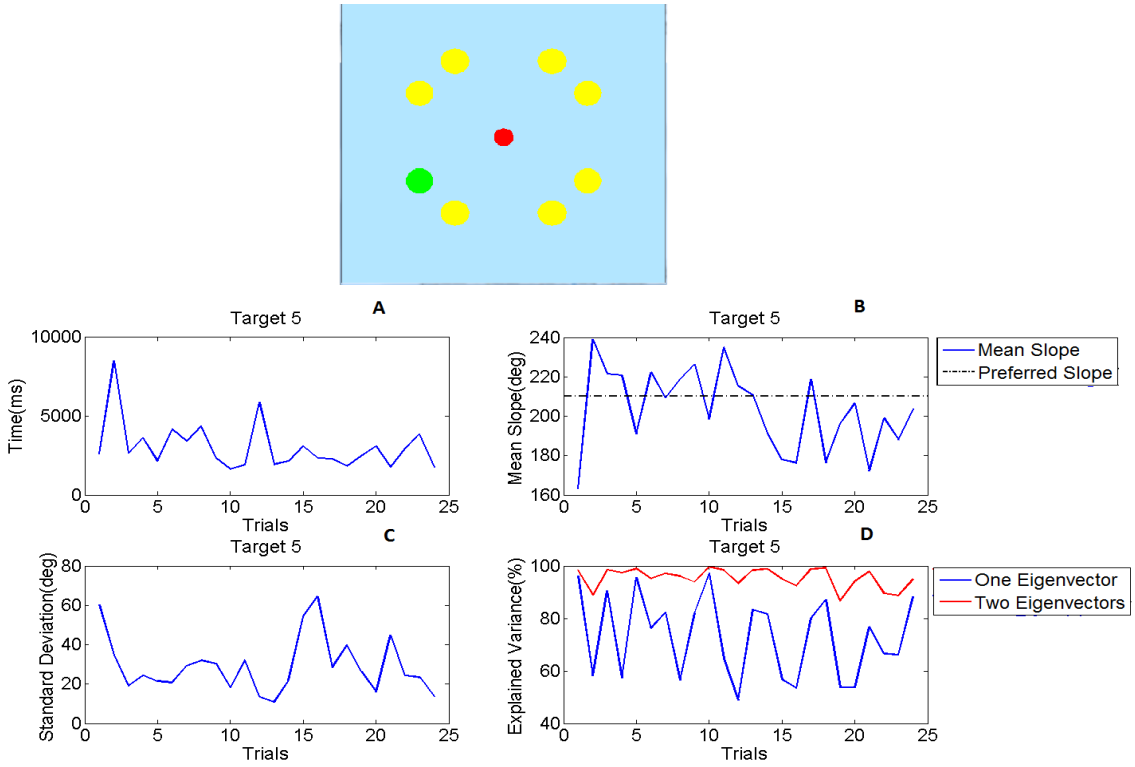

Figure 5.5: A: Time plot, B: Average slope plot, C: Standard deviation plot, D: Explained variance plot, for Target 5 (Experiment A)

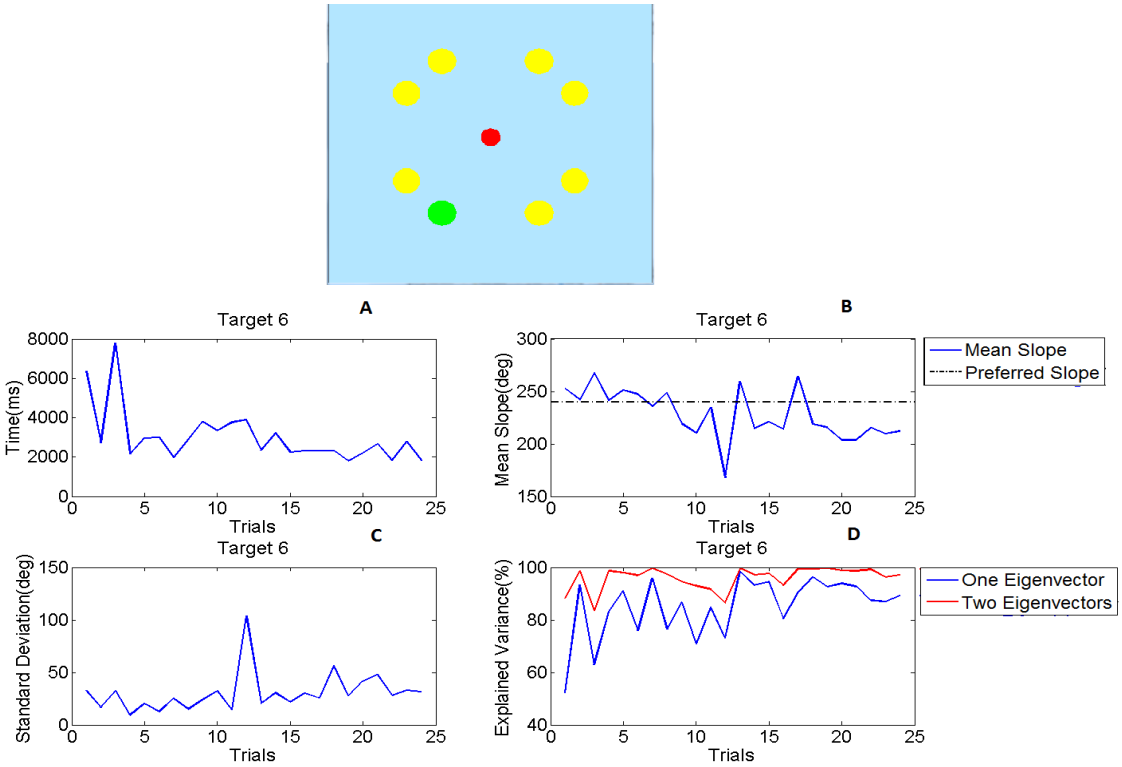

Figure 5.6: A: Time plot, B: Average slope plot, C: Standard deviation plot, D: Explained variance plot, for Target 6 (Experiment A)

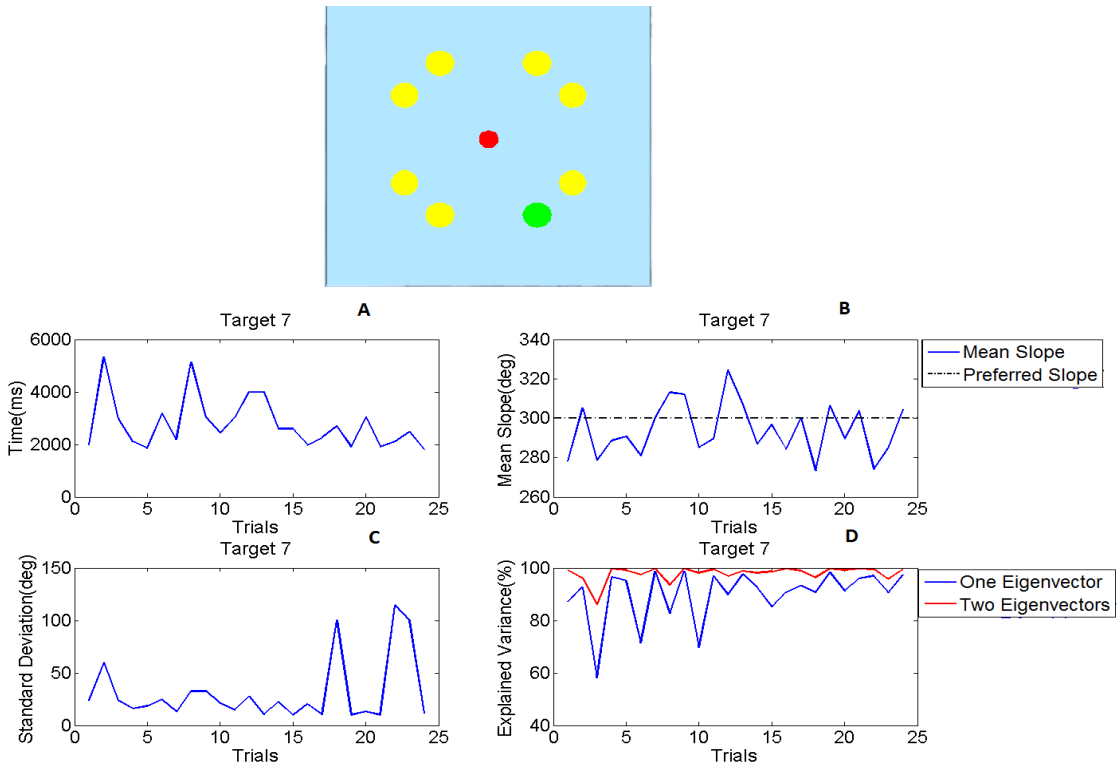

Figure 5.7: A: Time plot, B: Average slope plot, C: Standard deviation plot, D: Explained variance plot, for Target 7 (Experiment A)

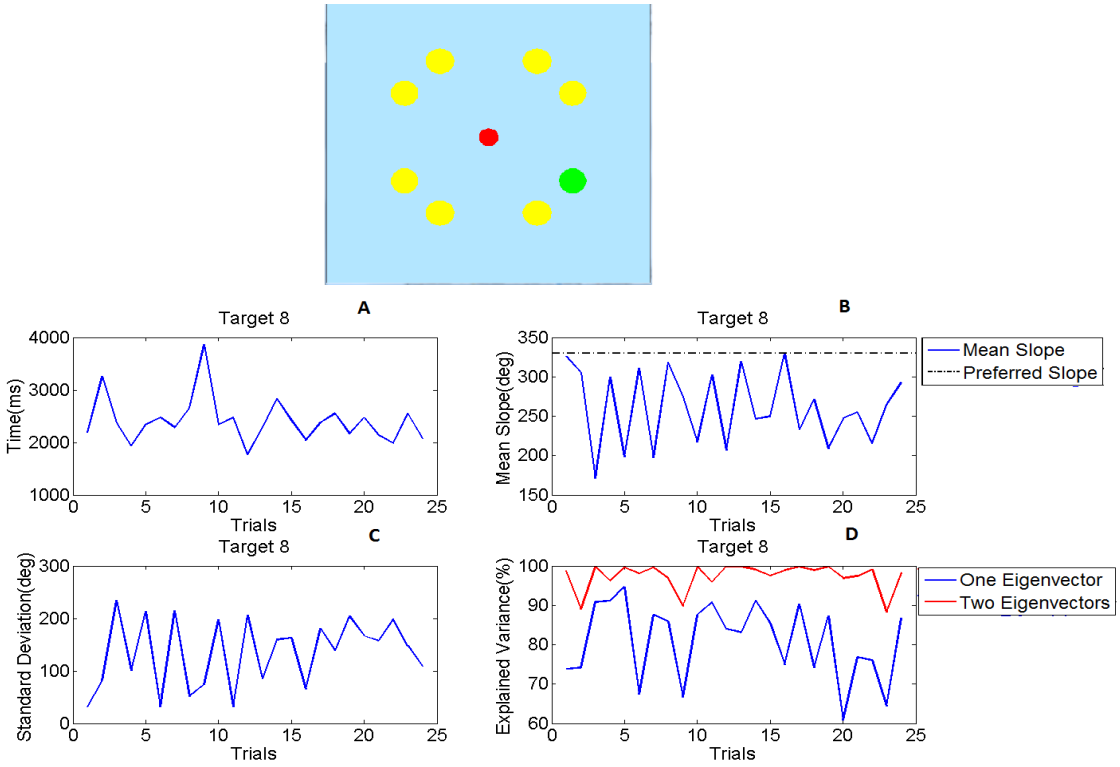

Figure 5.8: A: Time plot, B: Average slope plot, C: Standard deviation plot, D: Explained variance plot, for Target 8 (Experiment A)

Experiment B

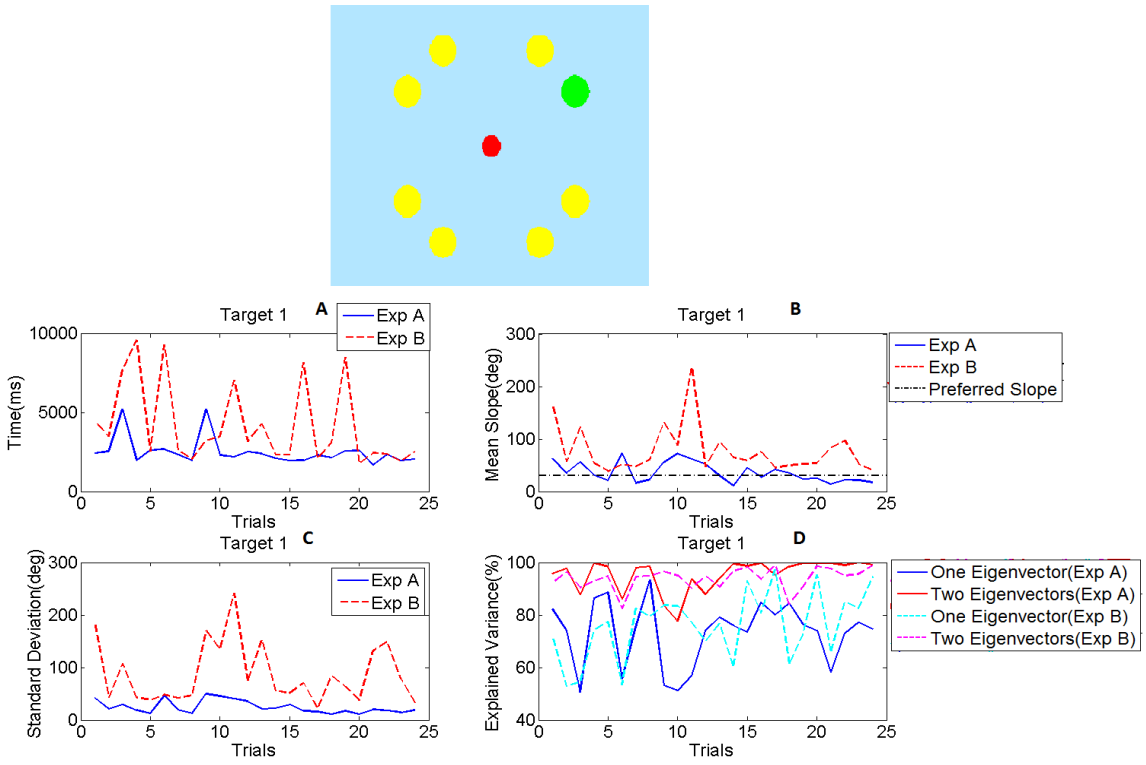

Figure 5.9: A: Time plot, B: Average slope plot, C: Standard deviation plot, D: Explained variance plot, for Target 1 (Experiment A and B)

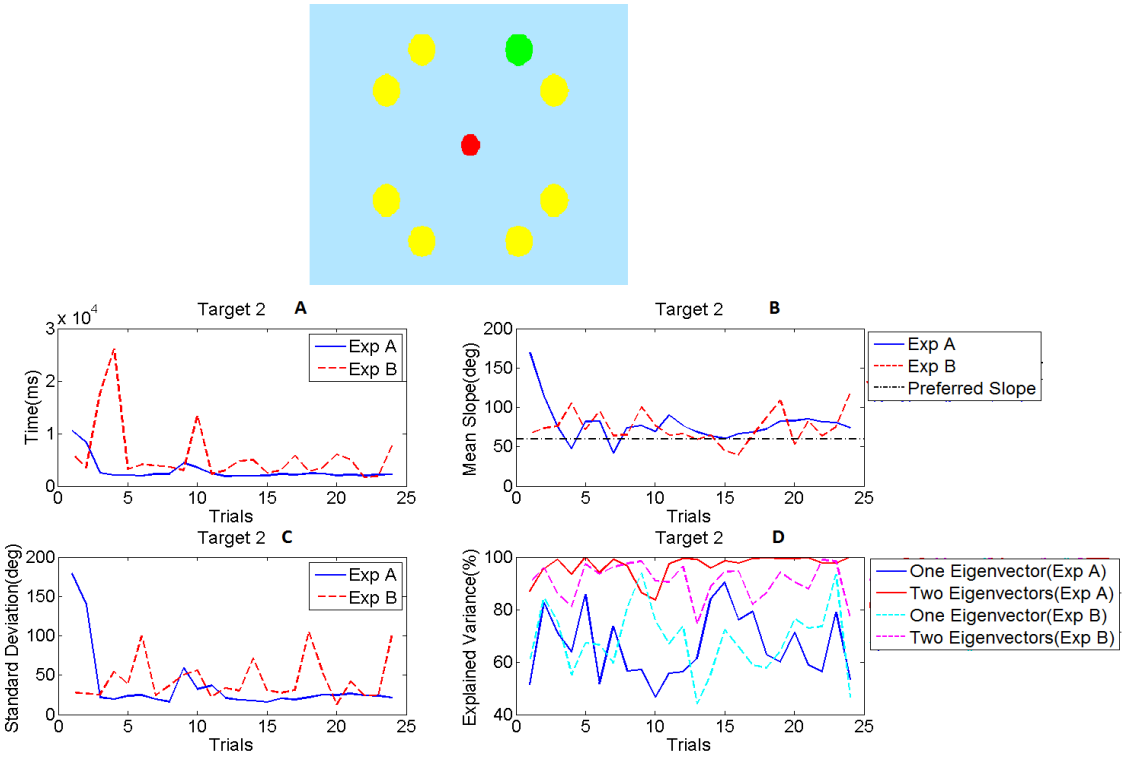

Figure 5.10: A: Time plot, B: Average slope plot, C: Standard deviation plot, D: Explained variance plot, for Target 2 (Experiment A and B)

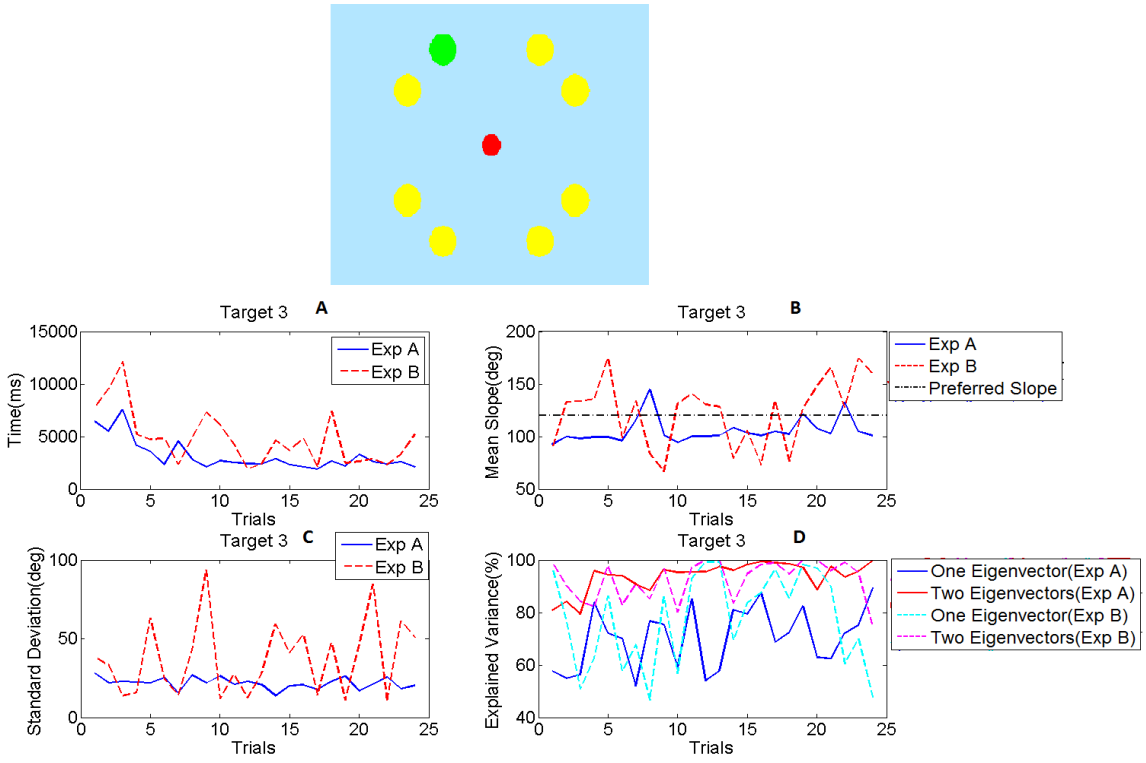

Figure 5.11: A: Time plot, B: Average slope plot, C: Standard deviation plot, D: Explained variance plot, for Target 3 (Experiment A and B)

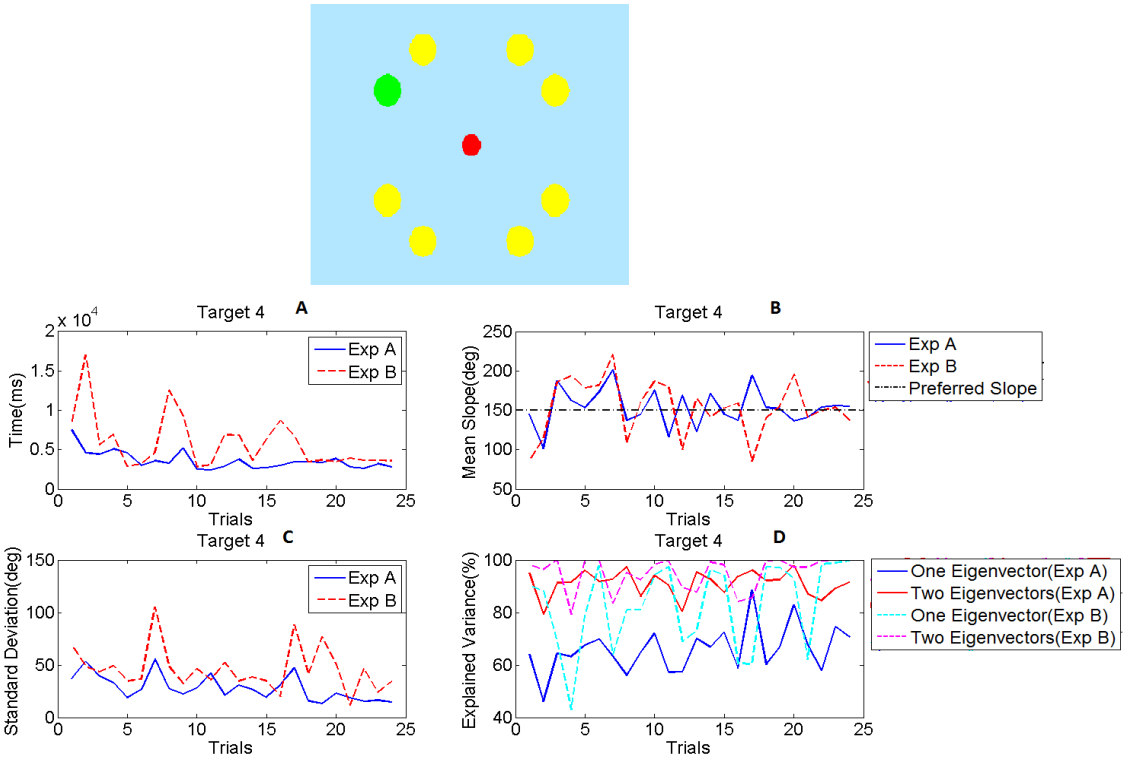

Figure 5.12: A: Time plot, B: Average slope plot, C: Standard deviation plot, D: Explained variance plot, for Target 4 (Experiment A and B)

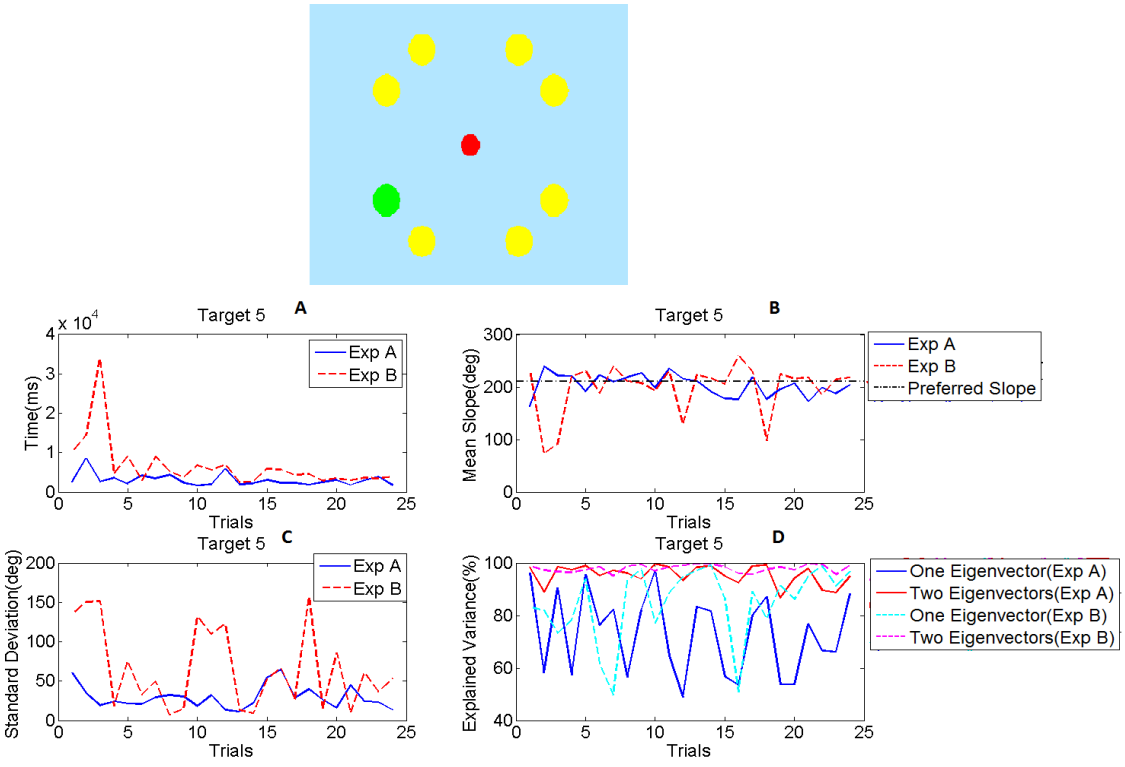

Figure 5.13: A: Time plot, B: Average slope plot, C: Standard deviation plot, D: Explained variance plot, for Target 5 (Experiment A and B)

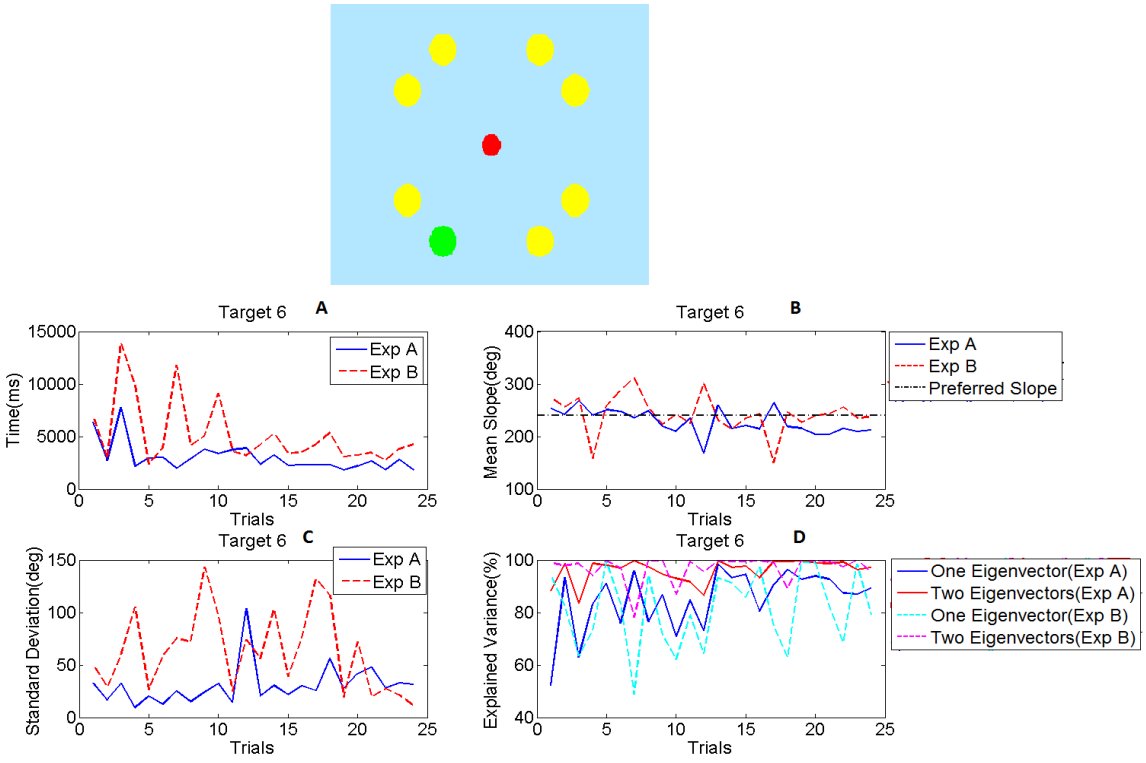

Figure 5.14: A: Time plot, B: Average slope plot, C: Standard deviation plot, D: Explained variance plot, for Target 6 (Experiment A and B)

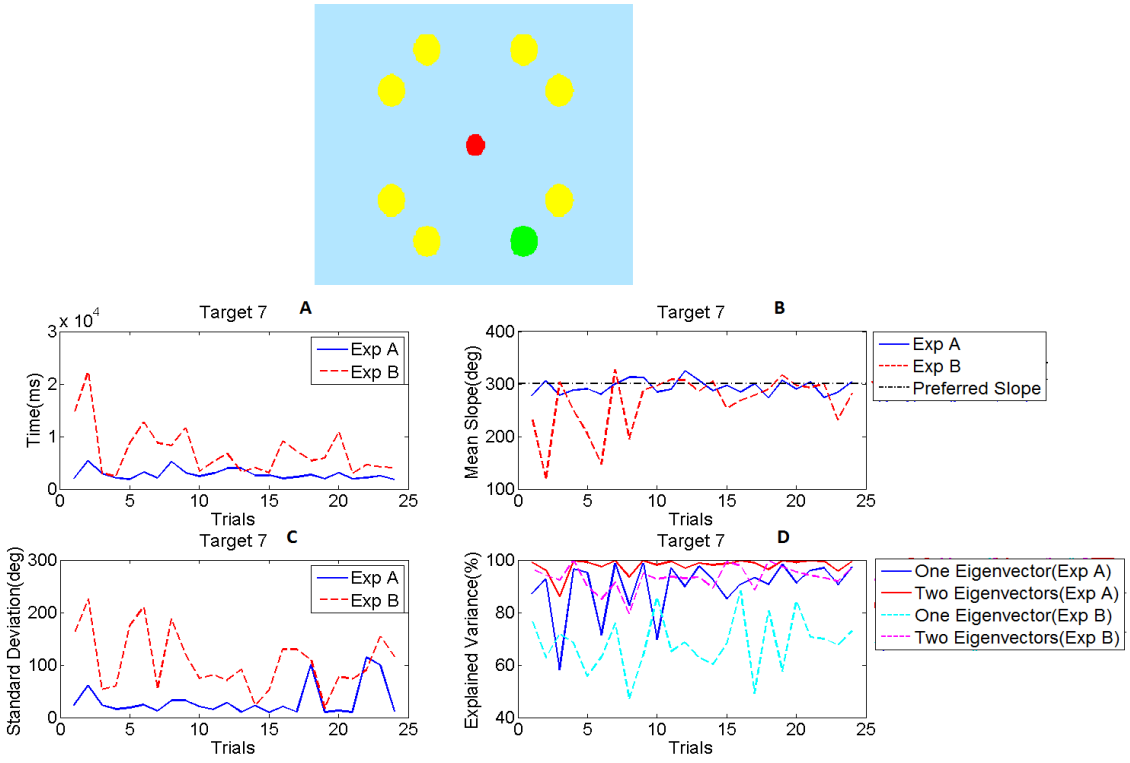

Figure 5.15: A: Time plot, B: Average slope plot, C: Standard deviation plot, D: Explained variance plot, for Target 7 (Experiment A and B)

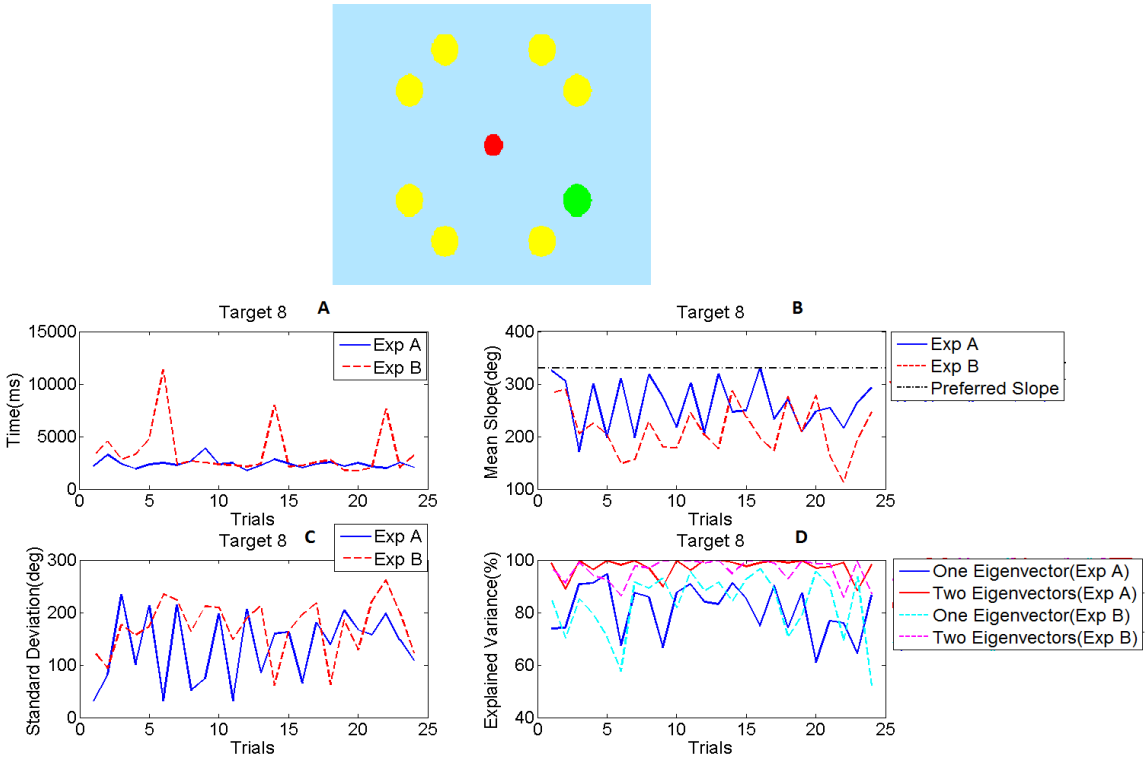

Figure 5.16: A: Time plot, B: Average slope plot, C: Standard deviation plot, D: Explained variance plot, for Target 8 (Experiment A and B)