Abstract—This document is written as final project for ELEC 548 course. I chose “Factor-Analysis Method for Higher-Performance Neural Prostheses” for implementation and performed two different Factor Analysis (FA) algorithms using data set “ReachData.mat”. Decoding Accuracy with the new version of FA algorithm described in the paper is about 70% while it is about 90% using the FA algorithm described in the text book. On the other hand, these values are reported about 95% and 90% respectively in the paper. There are several factors cause this discrepancy such as initialization and numerical errors.

I. ALGORITHM DESCRIPTION

The data sets in the paper are recorded from neural activities of two trained monkeys (H and G) performing a standard instructed-delay center-out reaching task. The data sets are recorded by implanting a 96-channel silicon electrode array in premotor cortex of the left hemisphere for monkey G and left hemisphere for monkey H under some special protocols and conditions. The window for counting neural spikes, known as “integration time” in the paper, starts 150ms after target presentation varies from 25ms to 250ms depending on analysis type. The datasets are divided into two parts named as “training trials” and “test trials”. The goal was to develop a system that is able to predict the reach target given only the neural data recorded from the implanted electrode array. The relationship between neural activity and reach point is exploited using a probabilistic framework. An initial set of training data is collected from both of the neural activity and reach endpoint. Training data is used for learning parameters for probabilistic model.

For each experimental trial, the square root number of observed spikes from each neural unit within integration window is stored into the vector \( \mathbf{y} \) and an abstract set of underlying factors known as “latent dimensions” are assembled into a vector \( \mathbf{x} \). where \( \mathbf{x} \) lies in a \( p \)-dimension (\( p \) is size of reduced data) space. These factors are Gaussian distributed, independent and have unit variance with mean 0, as shown in equation 1. Training data is passed to this probabilistic model per direction which causes 8 different learning parameters and 8 different probabilistic model. After learning fitting parameters per direction, test data is decoded by using the most probable target given the observed data. The complete model is as follows.

\[
P(\mathbf{x}) = N(\mathbf{x}; 0, I) \quad (1) \\
P(\mathbf{y}|\mathbf{x}, s) = N(\mathbf{y}; \mu_s + \mathbf{C}_s \mathbf{x}, \mathbf{R}_s) \quad (2) \\
P(s) = N(\mathbf{s}; \mu_s, \mathbf{C}_s \mathbf{C}_s^T + \mathbf{R}_s) \quad (3)
\]

In equation 2, \( \mu_s \) contains the mean number of spikes that each neuron would produce for endpoint \( s \) and is a \( q \) dimensional vector where \( q \) is the number of neuron units recorded in each trial. In fact, the assumption is there is only endpoint-related variability in the system and \( \mathbf{R}_s \) is a diagonal matrix that captures the independent variability present in \( \mathbf{y} \) form trial to trial. The matrix \( \mathbf{C}_s \) is a \( q \times p \) (\( q \) is original dimension of data and \( p \) is reduced dimension) matrix that does mapping between factors and observations. For the purpose of decoding, we can use equation 3 after learning all fitting parameters using FA algorithm. The mathematical derivation for FA algorithm is same as the text book. The FA algorithm is based on Expectation and Maximization (EM) technique and we can determine \( \mathbf{\mu}, \mathbf{C}, \) and \( \mathbf{R} \) in the factor analysis model by maximum likelihood. The solution for \( \mu \) is given by sample mean and \( \mathbf{C} \) must be found iteratively. Because factor analysis is a latent variable model, this can be done using EM algorithm. The equations for E step are as follows.

\[
E(\mathbf{x}_n) = \mathbf{G} \mathbf{C}^T \mathbf{R}^{-1}(\mathbf{y}_n - \bar{\mathbf{y}}) \quad (4) \\
E(\mathbf{x}_n | \mathbf{x}_n^T) = \mathbf{G} + E(\mathbf{x}_n)E(\mathbf{x}_n)^T \quad (5)
\]

Where \( \mathbf{G} \) is defined as:

\[
\mathbf{G} = (\mathbf{I} + \mathbf{C}^T \mathbf{R}^{-1} \mathbf{C})^{-1} \quad (6)
\]

The M step equations are reported below.

\[
\mathbf{C}^{\text{new}} = \frac{1}{N} \sum_{n=1}^{N} (\mathbf{y}_n - \mathbf{y}) \mathbf{E(} \mathbf{x}_n)\mathbf{E(} \mathbf{x}_n)^T \quad (7) \\
\mathbf{R}^{\text{new}} = \text{diag} \{ \mathbf{S} - \mathbf{C}^{\text{new}} \frac{1}{N} \sum_{n=1}^{N} \mathbf{E(} \mathbf{x}_n)(\mathbf{y}_n - \bar{\mathbf{y}})^T \} \quad (8)
\]

\( \mathbf{S} \) is covariance matrix of \( \{ \mathbf{x} \} \) and is calculated as equation 9.

\[
\mathbf{S} = \frac{1}{N} \sum_{n=1}^{N} (\mathbf{y}_n - \bar{\mathbf{y}})(\mathbf{y}_n - \bar{\mathbf{y}})^T \quad (9)
\]
In all of equations, N is total number of training trials which is set to be half of total number of trials in each direction.

There are 8 different sets of training data and each of them is related to a specific direction. By passing these 8 different data sets, we can extract 8 fitting parameter sets as $\mu$, $C$, and $R$ ($s \in \{1,2,...S\}$ and $S = 8$). In decoding phase, we calculate Probability of observed trial given fitting parameters and assign the observed trial to the direction with maximize the probability. Using equation 3 results the following equation for decoding the observed trial.

$$s = \text{argmax}_s \ P(y|s)$$

$$= \text{argmax}_s \ \frac{1}{|C_c^s + R|^2} \exp\left[-\frac{1}{2} (y - \mu_s)^T (C_c^s + R)^{-1} (y - \mu_s)\right]$$

The paper named this algorithm as “FAsep”.

The other method for FA algorithm reported in the paper is named “FAcmb”. In FAcmb approach, reach endpoint variability is not included in x vectors. However, reach endpoint could be simply another factor included in x vectors rather than being treated separately in the output space. In addition, FAcmb model requires computing a large number of parameters because there are different fitting parameters per direction. In order to alleviate these problems, FAcmb model is represented and formulated as follows.

$$P(x|s) = N(x; \mu_s, I)$$  \hspace{1cm} (11)

$$P(y|x) = N(y; Cx_n, R_s)$$  \hspace{1cm} (12)

$\mu_s$ is a p-dimensional vector describing the reach endpoint influence on the neural data within the lower-dimensional space. In contrast with FAsep method, C matrix is shared across reach endpoints unifying neural data. $R$ is a diagonal matrix and describes the independent variability similar to FAsep method. Because of shared mapping matrix $C$, output mapping is combined across reach targets and this method is “combined” version of Factor Analysis.

The E step equations are:

$$\xi_n = E[x_n|y_{n,s_n}, \theta^k]$$

$$= \mu_{s_n}^k + \beta_{s_n}^k (y_n - C \mu_{s_n}^k)$$  \hspace{1cm} (13)

$$\Sigma_n = E[x_n x_n^T|y_{n,s_n}, \theta^k]$$

$$= \Sigma_{s_n}^k - \beta_{s_n}^k C \Sigma_{s_n}^k + \xi_n^T \xi_n^T$$  \hspace{1cm} (14)

Where:

$$\beta_{s_n}^k = \Sigma_{s_n}^k C (R + C \Sigma_{s_n}^k C)^{-1}$$  \hspace{1cm} (15)

And $\Sigma_{s_n}$ is covariance matrix of latent variables.

The M step follows as:

$$N_s = \sum_{n=1}^N I(s_n = s)$$  \hspace{1cm} (15)

$$\mu_s^{k+1} = \frac{1}{N_s} \sum_{n=1}^N I(s_n = s) \xi_n$$  \hspace{1cm} (16)

$$\Sigma_{s_n}^k = \frac{1}{N_s} \sum_{n=1}^N I(s_n = s) \Sigma_n - \mu_{s_n}^{k+1} (\mu_{s_n}^{k+1})'$$  \hspace{1cm} (17)

$$C_{s_n}^{k+1} = (\Sigma_{s_n}^k)^{-1} (\sum_{n=1}^N \Sigma_n)^{-1}$$  \hspace{1cm} (18)

$$R^{k+1} = \frac{1}{N_s} \text{diag}(\sum_{n=1}^N y_n y_n') - C_{s_n}^{k+1} y_n y_n'$$  \hspace{1cm} (19)

“$I$” is indicator function and it would be equal to 1 when the trial direction is same as the direction that we are extracting parameters for it.

Decoding step is similar to FA separate algorithm and is done by using the equation below.

$$s = \text{argmax}_s \ P(y|s)$$

$$= \text{argmax}_s \ \frac{1}{|C_c^s + R|^2} \exp\left[-\frac{1}{2} (y - C \mu_s)^T (C_c^s + R)^{-1} (y - C \mu_s)\right]$$  \hspace{1cm} (20)

II. ALGORITHM IMPLEMENTATION AND RESULTS

I implemented both FAcmb and FAsep methods in MATLAB and tested the operation using data set “ReachData.mat”. It contains 1127 trials and 190 neural units. Similar to the paper, I stored square root of spike numbers within integration window which is 250ms and starts 150ms after planning start time. Trials are categorized based on intended direction and half of the trials per direction are selected randomly as training data and rest of them are used as test data. ReachData.mat is loaded and then by passing this data set to “spikedetector” function and square root of number of spikes within integration window is calculated for all trials. Then, half of trials in each direction are selected randomly and are passed to “FAsen” function. The number of iteration is set to be 100 in order to have higher accuracy. The output of this function includes mapping matrix, mean and independent variability matrix. For FAsep implementation, training data are passed to this function per direction and after passing all directions to this function, I have stored 8 different sets of fitting parameters. The next step is decoding phase, and direction of test trials is assigned using formula in equation (10). Decoding accuracy is defined as number of trails that are guessed correctly divided by total number of test data set trials.

The results of FAsep implementation is shown in figures (1) - (4).
The total decoding accuracy for this trial is 92.69%. FA<sub>cmh</sub> method is also implemented and tested using same data as in the last part. Instead of passing training data per direction to “CombinedFA” function, training data is passed at once and indicator function is calculated inside of “CombinedFA” function. Similar to FA<sub>sep</sub> method, the output of CombinedFA function includes fitting parameters but, there is only one mapping matrix and independent variability matrix for all directions and it outputs 8 different mean vectors for each direction. Similar to last part, for decoding, I passed test data to the equation 20. The results of this method implementation are shown in figure (3) and (4).

III. CONCLUSION

As is apparent from previous figures, decoding accuracy in FA<sub>sep</sub> method is higher than FA<sub>sep</sub> method while according to the paper results, I expected FA<sub>cmh</sub> to show higher decoding accuracy. There are some factors that may cause this discrepancy.

The number of neuron units for FA is not reported in the paper while I am using data set with 190 neural units; therefore, the data set used in the paper could be different from the data set that I used for implementation. Since, the output results is
dependent on input data, it can be one of the factors caused discrepancy. 
In addition, there are some neurons in the ReachData.mat data set that have 0 firing rates for almost all trials. It is equivalent of having a column with constant value which makes $\frac{1}{|c_i c_i' + R_s|^2}$ to have infinite value. To solve this problem, I assigned a random small number to spikecountsplan matrix. 
I also evaluated each method performance depending on the range of the assigned random numbers. $FA_{sep}$ had its best performance when the random number is a zero mean normal distribution with standard deviation 1 while $FA_{cmb}$ best performance corresponds to assigning a normal distributed random number with zero and standard deviation of 0.001. This fact releases that manipulating spike number data and assigning random number to 0 spikes, is another factor that caused the discrepancy. 
Moreover, the method for shuffling the data in the paper is different with the shuffling method that I used and it could be considered as another reason that the results are different.

Figure 6. Decoding Accuracy comparison for Separate and Combined algorithm for 20 runs.