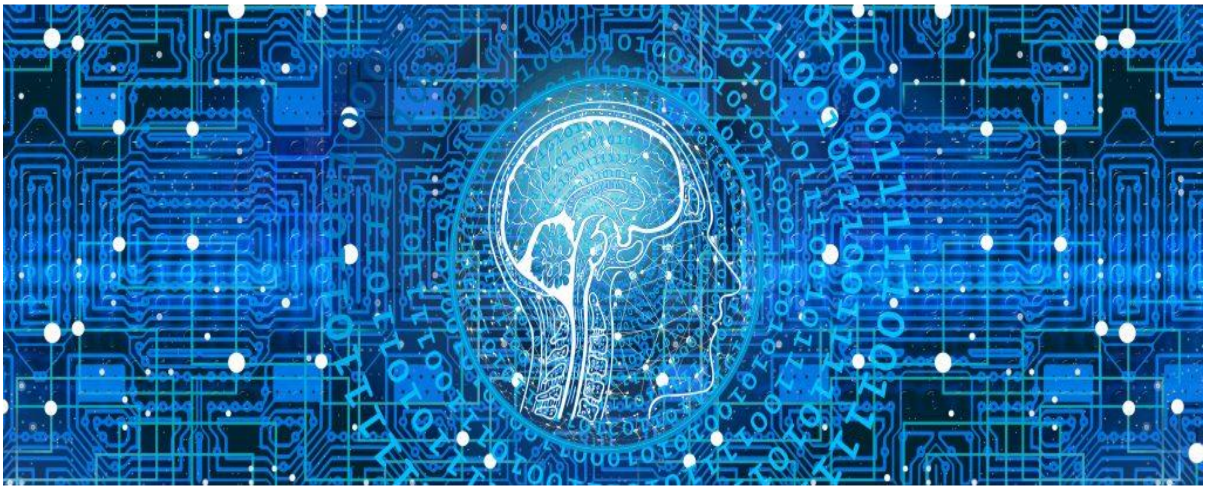
UNIVERSITY OF PATRAS
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING
MSc in Biomedical Engineering

APPLIED NEUROSCIENCE



Maria Revythi 1061099

A)Using tensor and/or matrix decomposition techniques, extract and plot the two main temporal components and the two main spatial components of EEG activity. Justify your model choice with reference to alternative techniques.

The matlab code I have used for tensor decomposition is introduced below:

```
%% tensor decomp
EEG=tensor(EEGData);

%% Apply PARAFAC
[P, U0] = cp_als(EEG,2);
figure;subplot(3,1,1)
bar(P.U{1}')
subplot(3,1,2)
plot(P.U{2})
subplot(3,1,3)
plot(P.U{3})

%% Apply Non-negative PARAFAC
[Pn, U0n] = cp_nmu(EEG,2);
figure;subplot(3,1,1)
bar(Pn.U{1}')
subplot(3,1,2)
plot(Pn.U{2})
subplot(3,1,3)
plot(Pn.U{3})

%% Apply Tucker
[Pt, U0t] = tucker_als(EEG,[2 2 2]);
figure;subplot(3,1,1)
bar(Pt.U{1}')
subplot(3,1,2)
plot(Pt.U{2})
subplot(3,1,3)
plot(Pt.U{3})

%% Apply Non-negative Tucker
opts=[];
[A,C,Out] = ntd(EEG,[2 2 2],opts)
figure;subplot(3,1,1)
bar(A{1,1}')
subplot(3,1,2)
plot(A{1,2})
subplot(3,1,3)
plot(A{1,3})
```

The results I have received after running this code are the following:

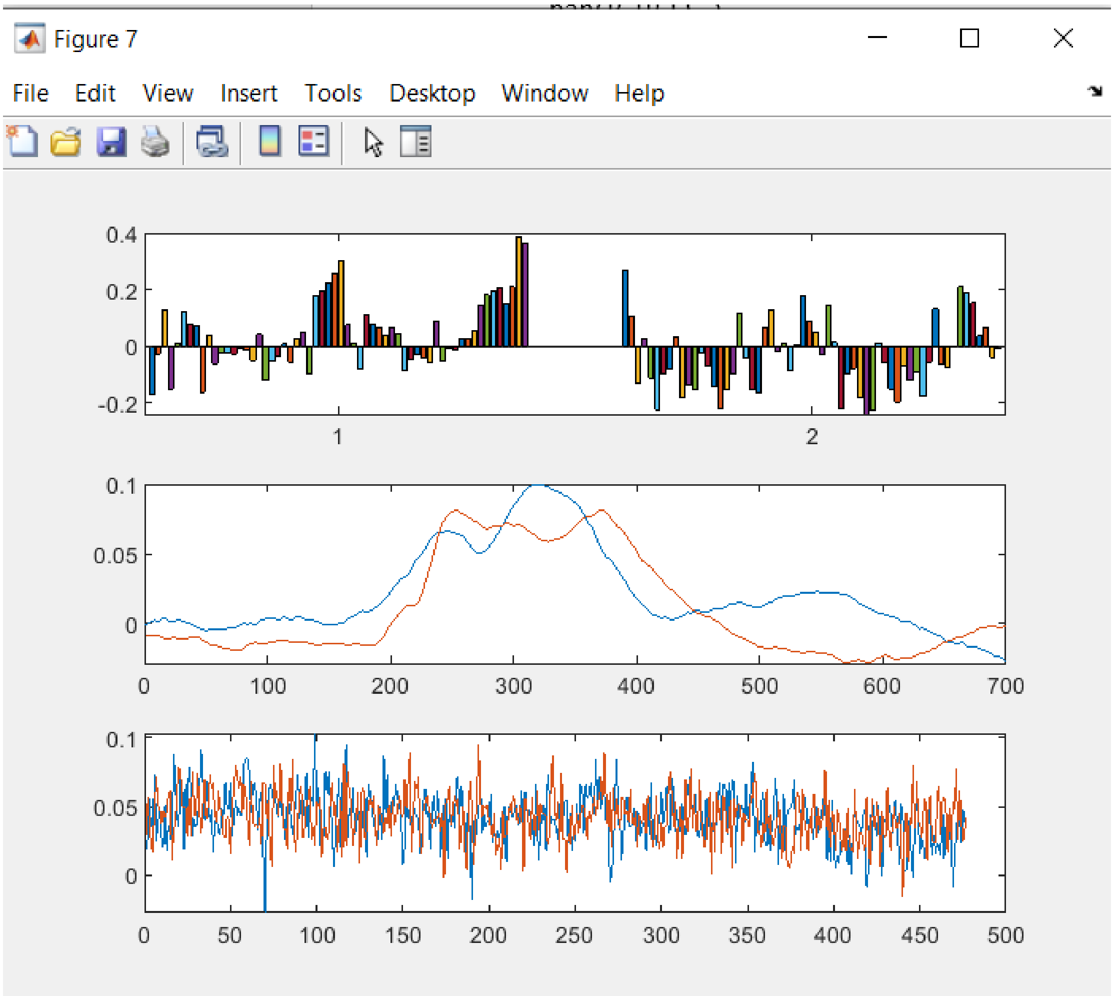


Figure 1: Spatial, temporal, trial components with parafac.

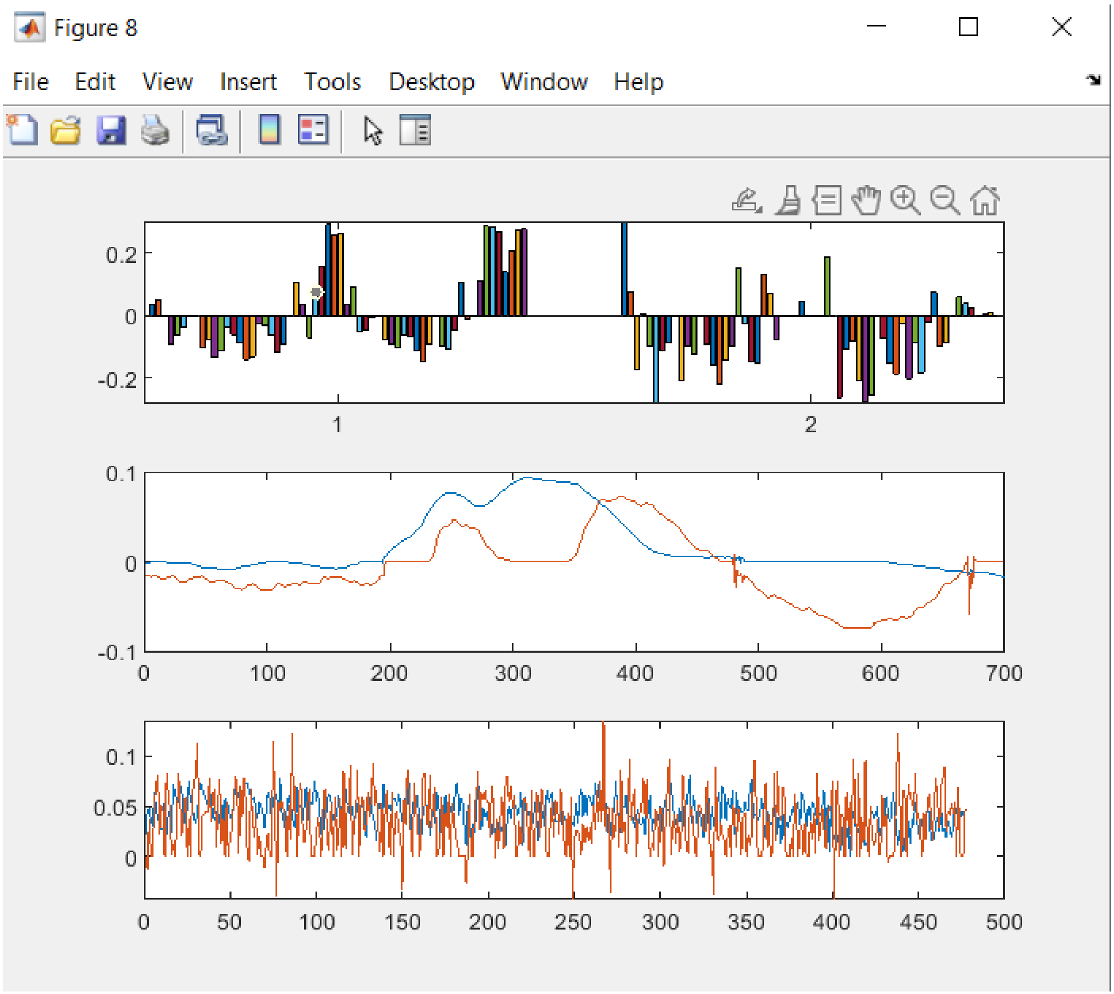


Figure 2: Spatial, temporal, trial components with non-negative parafac*.

*Maybe something is wrong because I have negative values in nonnegative parafac but I don’t know why.

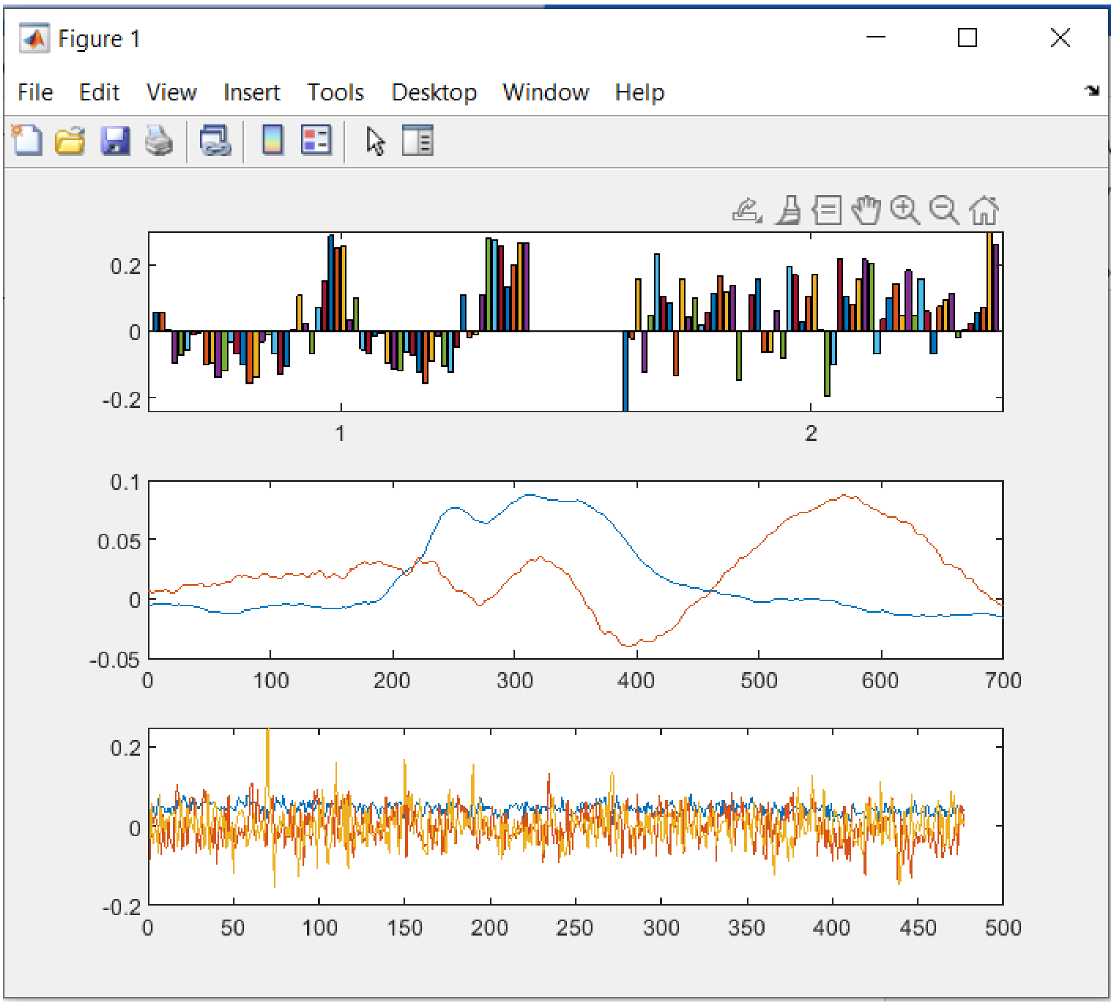


Figure 3: Spatial, temporal, trial components with Tucker.

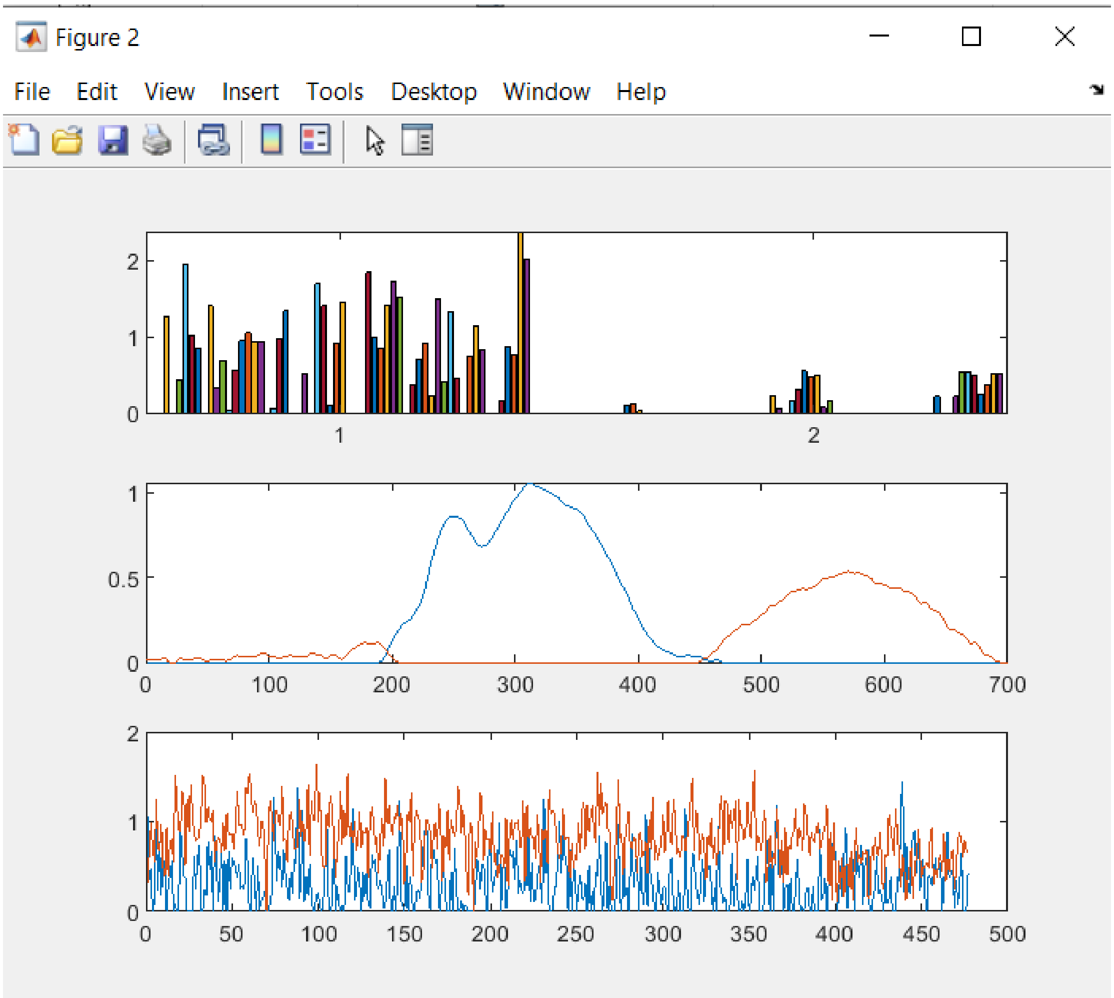


Figure 4: Spatial, temporal, trial components with non-negative Tucker.

In general, the major property of parafac is that I always know how i can combine the channels with the timepoints and the trials because of one-to-one correspondence. This indicates a very clear picture, which channels are activated in which timepoint and in which trial. However, tucker has not one to one correspondence because we can specify different number of components for channels, time and trials. In other words, tucker is more flexible. Furthermore, we can adapt one-to-one correspondence by using the same number of components in space ,time and trials. I think that the non-negative Tucker is the most appropriate because the spatial components are sparser, the time components do not overlap and there is no canceling effect in trial components.

Matrix decomposition techniques (pca, ica, nmf) are inappropriate for this data set, because we need the 3 dimensions to elaborate our data. With tensor decomposition we are being able to characterize the group of channels of temporal components separately but also how each group of channels is activated over time. I also know which trial each one of these components is active more. The data analysis is easier and faster with tensor decomposition. For these reasons, i preferred tensor decomposition. The matlab code I have used for matrix decomposition is introduced below:

```
%% temporal components
EEGt=permute(EEGData, [2 1 3]);
EEGt=reshape(EEGt, 700, 60*477);
size(EEGt)
%% PCA in time
[coefft,scoret,latentt,tsquaredt,explainedt,mut] = pca(EEGt');
figure;plot(coefft(:,1:2))

%% NMF in time
[Wt,Ht,Dt]=nnmf(EEGt,2);
figure;plot(Wt)

%% ICA in time
[Zt,Wicat,Tt]=fastICA(EEGt,2);
figure;plot(Wicat')

%% spatial components
EEGs=reshape(EEGData, 60, 700*477);
size(EEGs)
%% PCA  space
[coeff,score,latent,tsquared,explained,mu] = pca(EEGs');
figure;bar(coeff(:,1:2)')
%% NMF in space
[Wnmf,H,D]=nnmf(EEGs,2);
figure;bar(Wnmf')
%% ICA in space
[Z,Wica,T]=fastICA(EEGs,2);
figure;bar(Wica)
```

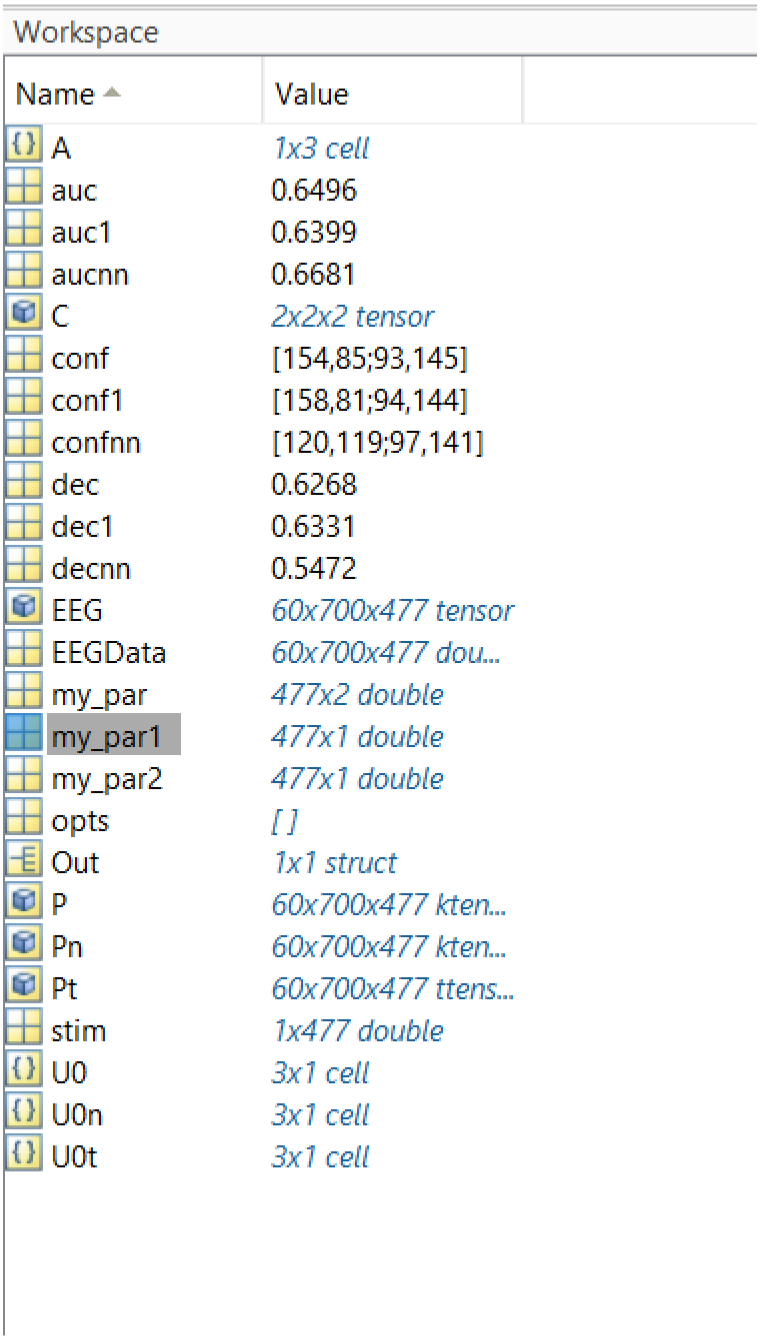
However, as i described before, the results were not satisfying.

B)   Use the extracted EEG components to discriminate which stimulus was presented on each trial. What is the discrimination accuracy? Can you improve this by using a different classifier?

The discrimination accuracy for a discriminant linear classifier is 0.6268. For binary tasks the accuracy must be above 0.5 and more close to 1.

The higher it is the better classification I have.

They are different ways of selecting classifiers and this is depending on the data. So, a different classifier may give more meaningful results.



Workspace

Name	Value
A	1x3 cell
auc	0.6496
auc1	0.6399
aucnn	0.6681
C	2x2x2 tensor
conf	[154,85;93,145]
conf1	[158,81;94,144]
confnn	[120,119;97,141]
dec	0.6268
dec1	0.6331
decnn	0.5472
EEG	60x700x477 tensor
EEGData	60x700x477 dou...
my_par	477x2 double
my_par1	477x1 double
my_par2	477x1 double
opts	[ ]
Out	1x1 struct
P	60x700x477 kten...
Pn	60x700x477 kten...
Pt	60x700x477 ttens...
stim	1x477 double
U0	3x1 cell
U0n	3x1 cell
U0t	3x1 cell

Figure 5: My matlab workspace.

The variable auc1, dec1 corresponds to quadratic classifier.

The variable auc, dec corresponds to linear classifier.

The variable aucnn, decnn corresponds to nearest neighbor classifier.

In comparison with the 3 different classifiers I have used, i think that the linear classifier is the best choice. However, they are an infinite number of different classifiers, so another classifier may give results more closer to 1.

Which component or combination of components carry most of the discrimination power? Illustrate this by plotting i) the corresponding confusion matrices and ii) scatter plot(s) of the component activations.

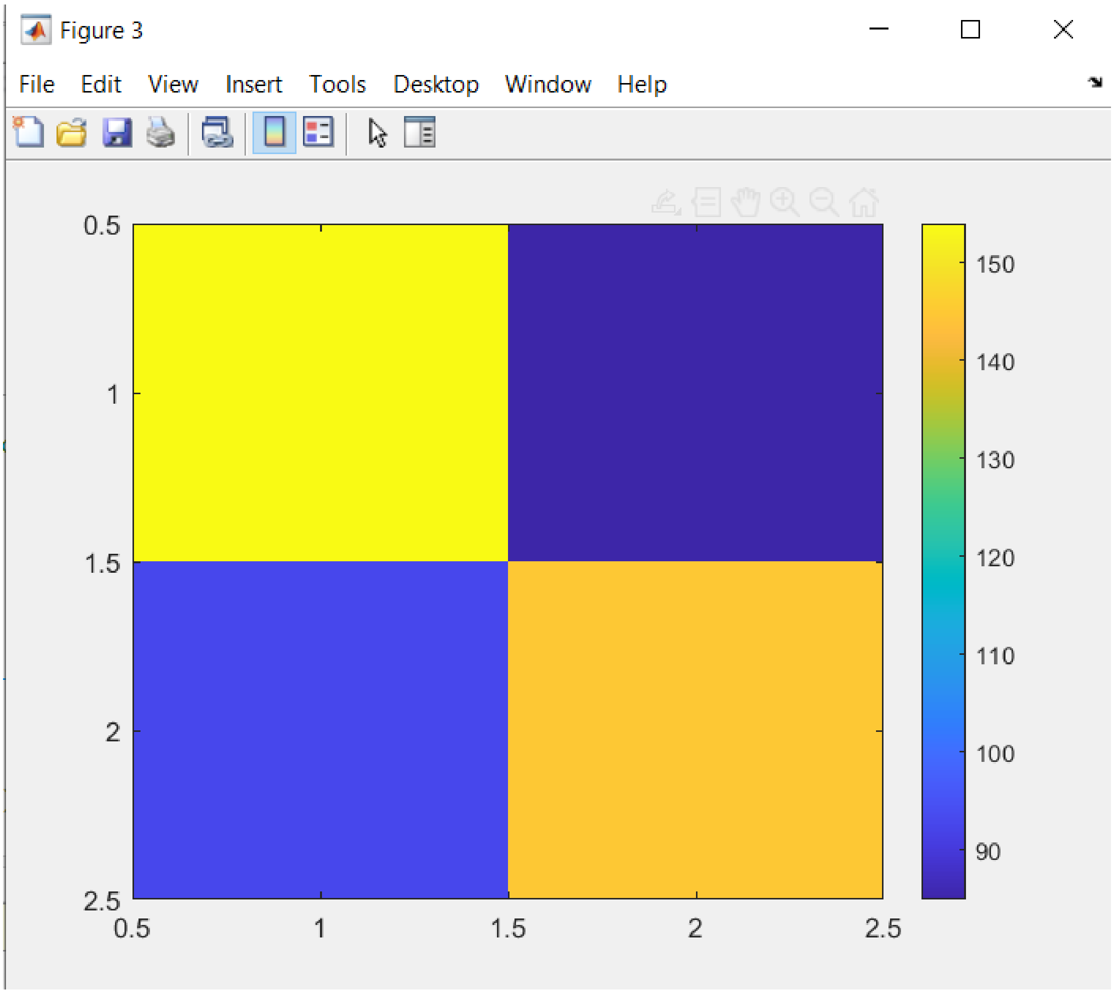


Figure 6: Corresponding confusion matrix of first and second component.

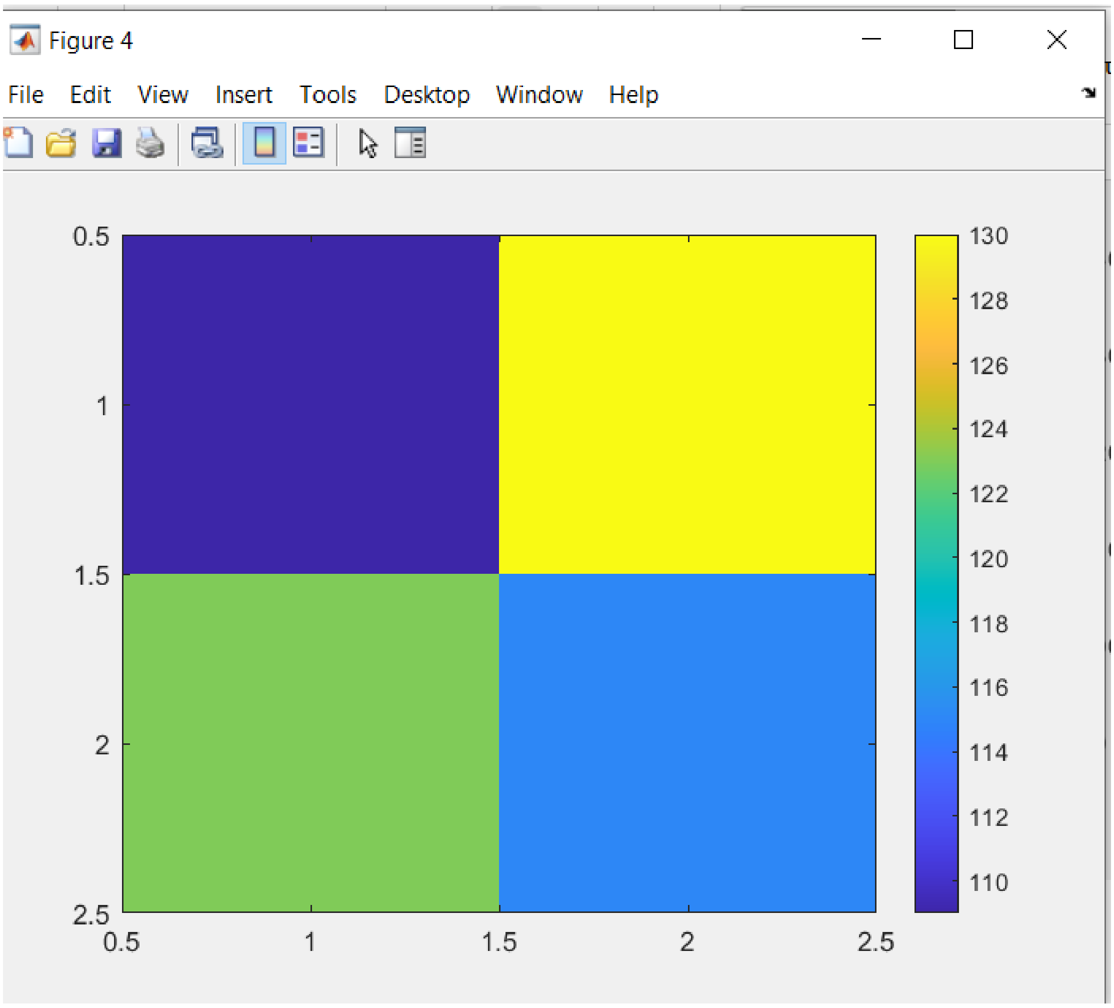


Figure 7: Corresponding confusion matrix of first component.

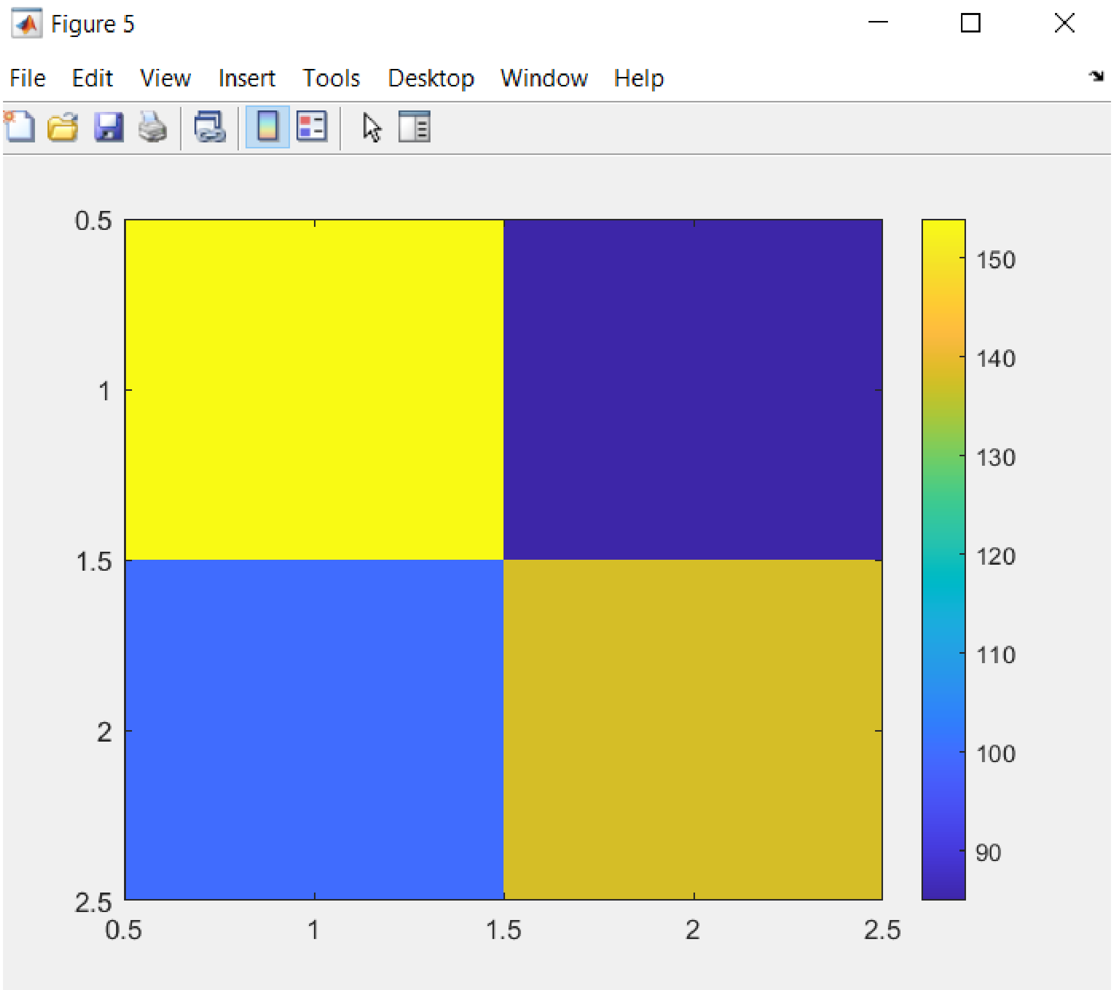


Figure 8: Corresponding confusion matrix of second component.

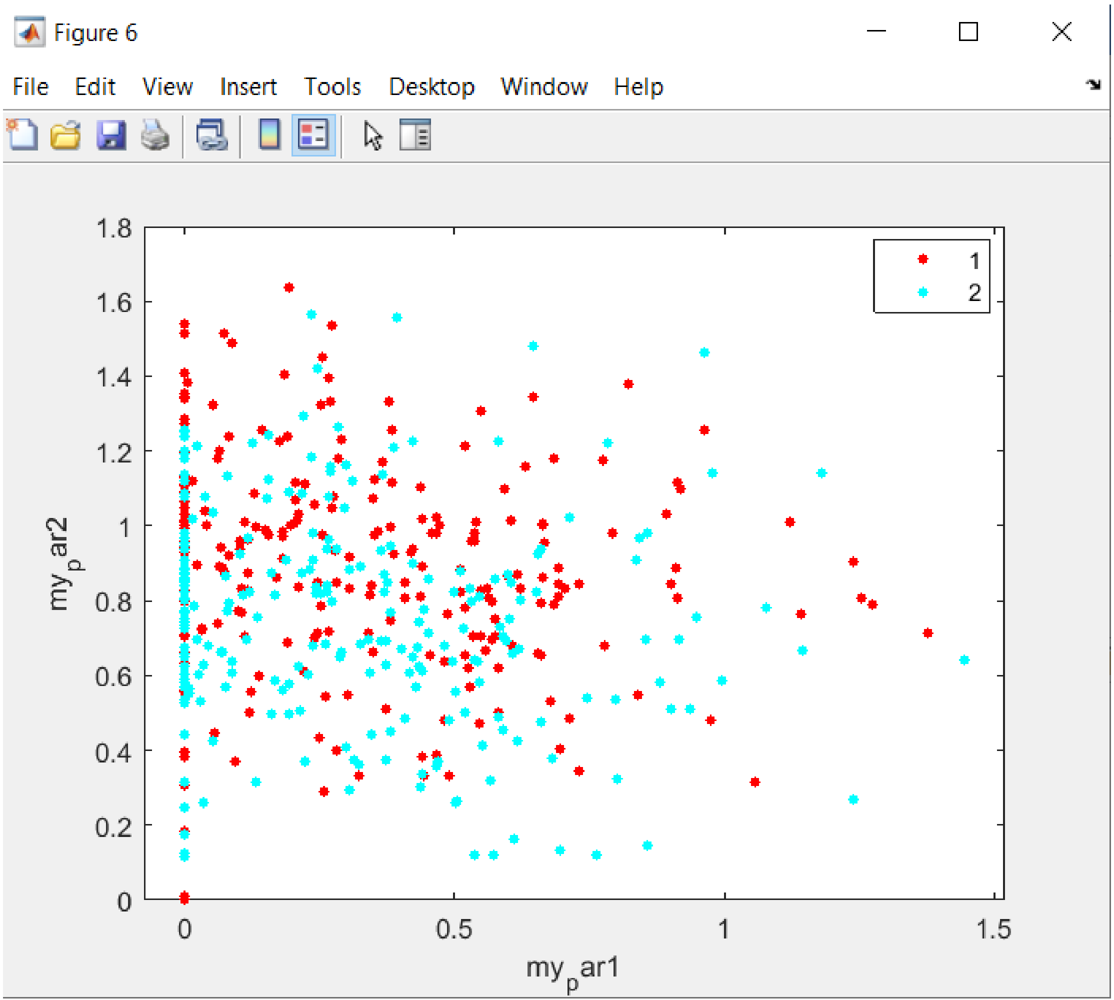


Figure 9: Scatter plot.

The confusion matrix includes more information than how well I decode. A confusion matrix includes information of which component I classify well and which component I don't classify well, where my errors are and how often I make these errors. In general, the more I have on the diagonal the better classification I get.

The biggest values on diagonal I have on figure 6 so the 2 components act complementary. However, during the comparison between figure 7 and 8, I can conclude that component 2 is more informative than component 1.

A scatter plot can inform the efficiency of the classification. If a linear boundary can be used to discriminate the red from the blue dots, then the classification is good. In this case it is difficult to use a linear boundary, as a result the classification isn't satisfactory.

The matlab code, which I have used for the classification, is the following:

```
%%   Non-negative Tucker parameters
```

```
my_par1=A{1,3}(:,1);   %1= my first compoment
my_par2=A{1,3}(:,2);

my_par=[my_par1 my_par2];
% classify stim
[auc,dec,conf]=decoder(my_par,stim');
figure;imagesc(conf);colorbar
%% scatterplots
 %figure;gscatter(my_par1,my_par1,stim)
 figure;gscatter(my_par1,my_par2,stim)
 figure;gscatter (A{1,3}(:,1),A{1,3}(:,2),stim)
```

Do all combinations of spatial and temporal components contribute to stimulus discrimination? Based on your answer, explain the functional roles of the extracted components.

I think that all components complement each other. The second component is more informative about the discrimination task than the first component. The first spatial component is more informative about the location of the channels.