ASSIGNMENT 4

-Meghana Venkata Palukuri

-CH12B083

Question 1:

The goal is to extract the pure component spectra from the mixture UV absorbance data set used in assignment 3. We assume that the number of pure components in the mixture is known to be three. PCA is used to first obtain the denoised absorbance matrix of rank three and then NMF is applied to the denoised data to extract the three sources. Using correlation coefficients for extracted and actual spectra, we identify of the extracted spectra. The shape of the absorbance spectra of the extracted spectra is also compared with the actual ones.

This is done for two cases- random sampling of the replicates for each mixture and averaged value of the replicates for each mixture.

```
clc
clear all
close all

Data=load('Inorfull.mat');
% Data structure consists of the follow attributes:
%            CONC: [130x3 double]
%            DATA: [130x176 double]
%          PureCo: [1x176 double]
%          PureCr: [1x176 double]
%          PureNi: [1x176 double]
%             WAV: [1x176 double]
%         stdDATA: [130x176 double]
%      PureCoCONC: 0.1720
%      PureCrCONC: 0.0764
%      PureNiCONC: 0.1965
```

a) Random replicate

```
close all
NewData=ones(26,176);
NewStdData=ones(26,176);
NewConc=ones(26,3);

rand('seed',95)
for i=1:26
    dt_num = randi(5);
    istart = 5*(i-1)+dt_num;
    NewStdData = [NewStdData; Data.stdDATA(istart,:)];
     NewData = [NewData; Data.DATA(istart,:)];
    NewConc = [NewConc;Data.CONC(istart,:)];
end

% Final Data: Absorbance spectra (176 pts.) measured for each sample
% (Total: 26 samples). , each consisting of 3 components with their
% concentrations noted.

PureComp=[Data.PureCo; Data.PureCr; Data.PureNi];

 ModelY= NewConc; %Conc is Y since goal is to predict concentration given absorbance spectra
 %Linv = inv(diag(mean(NewStdData)));
 %ModelX=NewData*Linv;
ModelX= NewData; %Unscaled Data
```

```
%Denoising the data using PCA
DenoisedModelData = PCA(ModelX,3);
%DenoisedModelData = (DenoisedModelData')*(diag(mean(NewStdData)));
k=3;
```

```
% Applying NMF to extract the 3 sources. NMF function by Prof. Haesun Park
[W,H,iter,HIS]=nmf(DenoisedModelData’,k);

% Finding correlation coefficients to determine which component it corresponds to


correlation_coeff = zeros(3,3);
for i = 1:3
    for j = 1:3
        correlation_coeff(i,j) = corr2(PureComp(i,:),H(j,:));
    end
end

correlation_coeff_rand=correlation_coeff
PureCompPred=[H(3,:) ; H(1,:); H(2,:)]; % The extracted pure component spectras are in this
%order obtained from the correlation coefficients
```

```
%Plotting the spectras
figure(1)
plot(300:2:650,PureComp(1,:),'b-')
title('Pure Co Spectrum')

figure(2)
plot(300:2:650,PureCompPred(1,:),'g-')
title('Predicted Pure Co Spectrum')

figure(3)
plot(300:2:650,PureComp(2,:),'b-')
title('Pure Cr spectrum')

figure(4)
plot(300:2:650,PureCompPred(2,:),'g-')
title('Predicted Pure Cr spectrum')

figure(5)
plot(300:2:650,PureComp(3,:),'b-')
title('Pure Ni spectrum')

figure(6)
plot(300:2:650,PureCompPred(3,:),'g-')
```

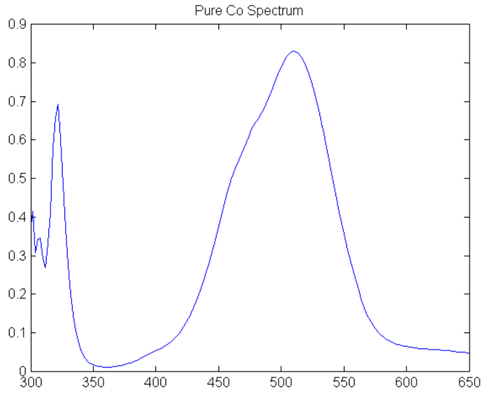
```
title('Predicted Pure Ni spectrum')
```

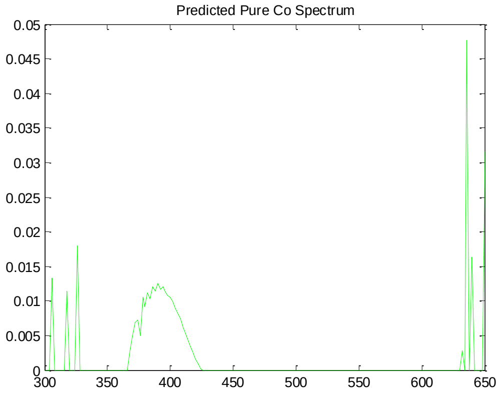
```
par = 

              m: 52
              n: 176
           type: 'regularized'
    nnls_solver: 'bp'
          alpha: -1.5488e-04
           beta: -1.5488e-04
       max_iter: 100
       min_iter: 20
       max_time: 1000000
            tol: 1.0000e-03
        verbose: 0


final = 

        iterations: 23
     elapsed_total: 0.0938
    relative_error: 0.7253
         W_density: 0.5769
         H_density: 0.4451
```

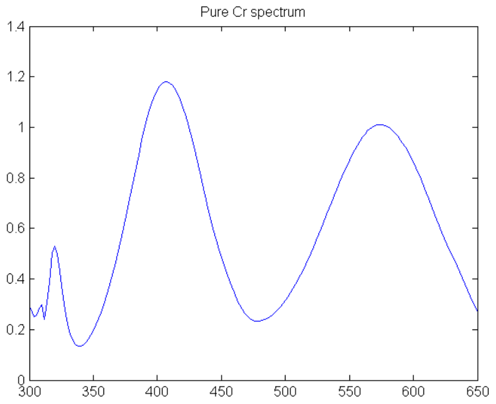
```
correlation_coeff_rand =

   -0.3283   -0.3402   -0.2529

    0.5751    0.4485    0.1529

    0.8525    0.8501    0.5459
```

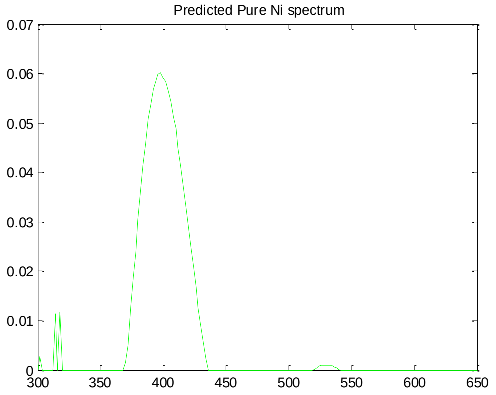
* Rows are actual pure components and columns are predicted/extracted pure components.

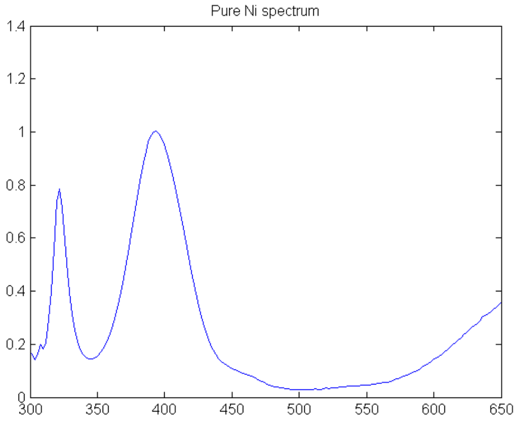
The mapping is chosen in such a way that there is maximum correlation between the combination of the mapped values (since in this case 2 values (corresponding to True Cr and Ni spectra) show maximum correlation to the same mapped quantity, the 2nd largest is chosen as a trade-off. The result is justified when we plot the graphs).

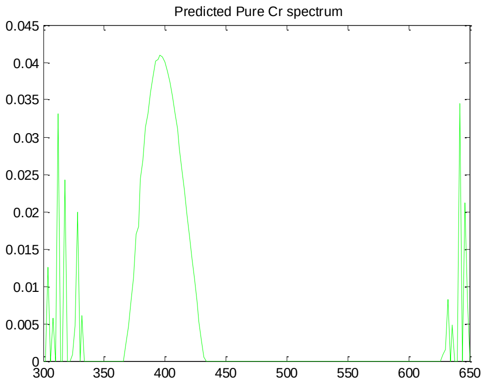












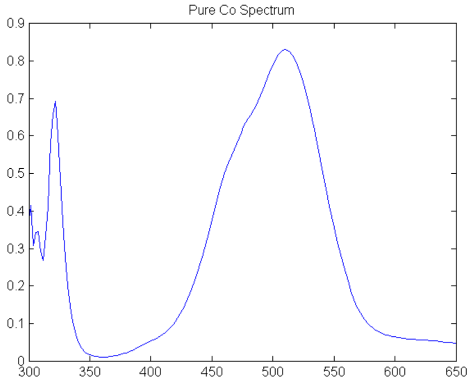
b) With Average of the replicates

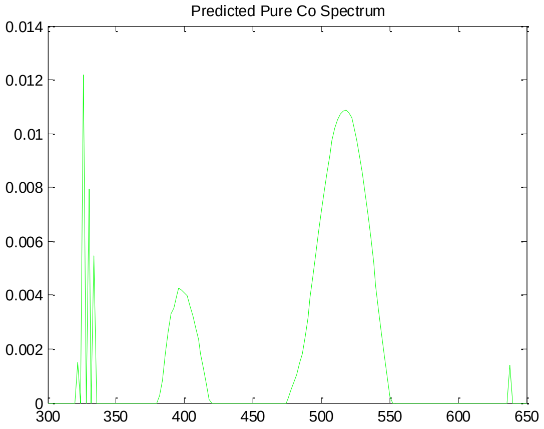
```
% Average measurement
NewData=ones(26,176);
NewStdData=ones(26,176);
NewConc=ones(26,3);

count=1;
for i=1:5:130

NewConc(count,:)=(Data.CONC(i,:)+Data.CONC(i+1,:)+Data.CONC(i+2,:)+Data.CONC(i+3,:)+Data.CONC
(i+4,:))/5;

NewData(count,:)=(Data.DATA(i,:)+Data.DATA(i+1,:)+Data.DATA(i+2,:)+Data.DATA(i+3,:)+Data.DATA
(i+4,:))/5;

NewStdData(count,:)=(Data.stdDATA(i,:)+Data.stdDATA(i+1,:)+Data.stdDATA(i+2,:)+Data.stdDATA(i
+3,:)+Data.stdDATA(i+4,:))/5;
    count=count+1;
end

PureComp=[Data.PureCo; Data.PureCr; Data.PureNi];

 ModelY= NewConc;
 %Linv = inv(diag(mean(NewStdData)));
 %ModelX=NewData*Linv;
ModelX= NewData;

%Denoising the data using PCA
DenoisedModelData = PCA(ModelX,3);
%DenoisedModelData = (DenoisedModelData')*(diag(mean(NewStdData)));
k=3;
```

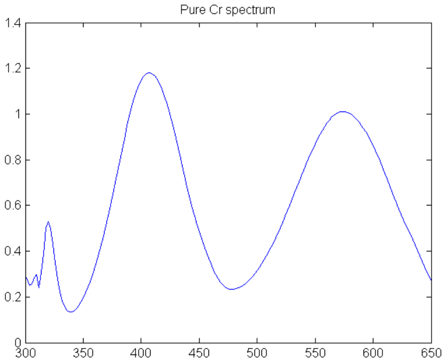
```
% Applying NMF to extract the 3 sources
[W,H,iter,HIS]=nmf(DenoisedModelData’,k);

% Finding correlation coefficients to determine which component it corresponds to

correlation_coeff = zeros(3,3);
for i = 1:3
    for j = 1:3
        correlation_coeff(i,j) = corr2(PureComp(i,:),H(j,:));
    end
end

correlation_coeff_avg=correlation_coeff
PureCompPred=[H(3,:) ; H(2,:); H(1,:)];

figure(1)
plot(300:2:650,PureComp(1,:),'b-')
title('Pure Co Spectrum')

figure(2)
plot(300:2:650,PureCompPred(1,:),'g-')
title('Predicted Pure Co Spectrum')

figure(3)
plot(300:2:650,PureComp(2,:),'b-')
title('Pure Cr spectrum')

figure(4)
plot(300:2:650,PureCompPred(2,:),'g-')
title('Predicted Pure Cr spectrum')

figure(5)
plot(300:2:650,PureComp(3,:),'b-')
title('Pure Ni spectrum')

figure(6)
plot(300:2:650,PureCompPred(3,:),'g-')
title('Predicted Pure Ni spectrum')
```

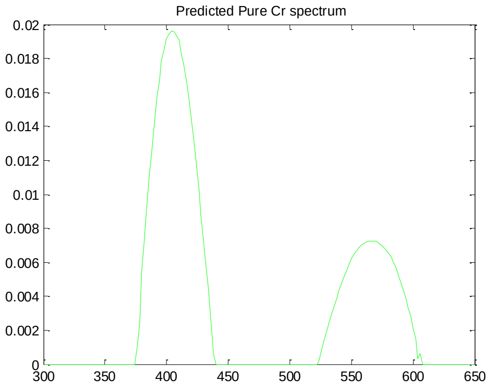
```
par = 

              m: 26
              n: 176
           type: 'regularized'
    nnls_solver: 'bp'
          alpha: -0.1155
           beta: -0.1155
       max_iter: 100
```

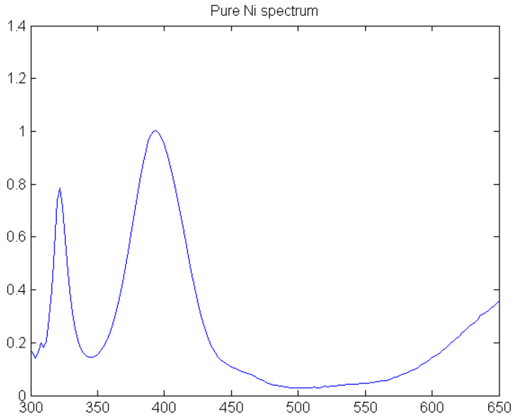
```
        min_iter: 20
        max_time: 1000000
             tol: 1.0000e-03
         verbose: 0


final =

           iterations: 23
        elapsed_total: 0.0313
       relative_error: 0.9998
            W_density: 1
            H_density: 0.6061


correlation_coeff_avg =

   -0.3459   -0.3522    0.6360

    0.5571    0.8255   -0.0758

    0.8663    0.6134   -0.0814
* Rows are actual pure components and columns are predicted/extracted pure components
```

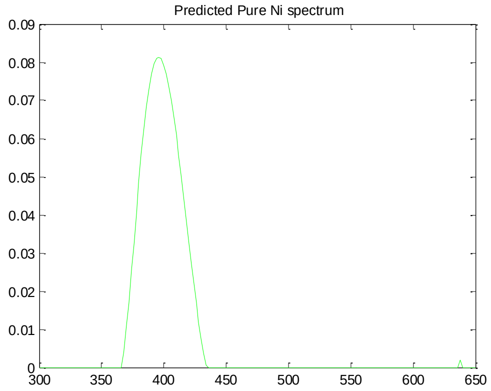










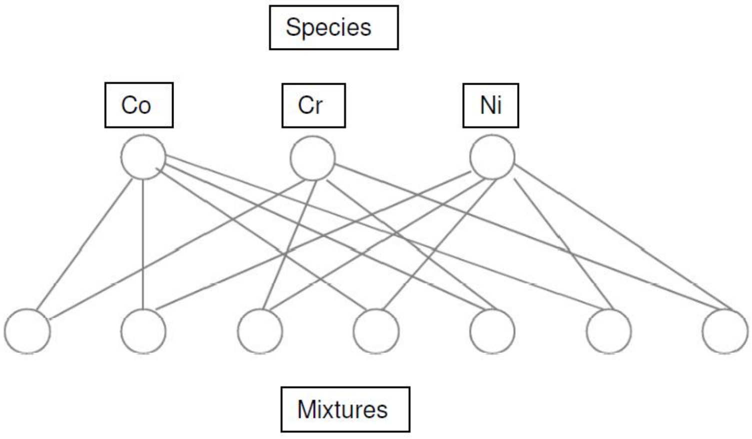


CONCLUSIONS:

From the correlation coefficients as well as the graphs we can compare the extracted component spectra to their corresponding actual spectra (for the unscaled data). The positions of the peaks, number of peaks, peak heights and relative peak heights are captured with varying degrees of accuracy, with the average case showing better correspondence to the true spectras. The average case has much better correlation coefficients for each of the components when compared to the random case. The present random case is in fact erroneous in predicting Co spectrum as can be seen from the very low correlation coefficient as well as the graph. Therefore the average case is better, as the effect of erroneous measurements is reduced.

Question 2:

UV absorbances for seven 3-component mixtures obtained by preparing mixtures consisting of Co, Cr, and Ni salts in nitric acid according to the experimental design are given. The pure component spectra are extracted by the following 2 methods. PCA is applied to the data set and denoised spectra of mixtures is obtained. The rotation matrix is determined using the experimental design information and applied to the denoised spectra to estimate the pure component spectra. The second method is to apply NCA. The extracted spectra is compared through correlation coefficients and graphs.



```
clc
clear all
close all
```

a) PCA

```
Data=load('ncadata.mat');

ModelData=Data.measabs;
PureComp=Data.pureabs;
%Denoising the data using PCA
DenoisedModelData = PCA(ModelData,3);

% Rotation Matrix based on connectivity 7*3 matrix in the order of Co, Cr,
% Ni
ConnectivityMatrix=[ 1 1 0 ; 1 0 1; 0 1 1; 1 0 1; 1 1 0; 1 0 1; 0 1 1];
RotMat=ConnectivityMatrix
PureCompPred=(DenoisedModelData*RotMat)';


% Calculating RMSE
RMSE=0;
for i=1:3
    for j=1:176
RMSE=RMSE+(PureComp(i,j)-PureCompPred(i,j))^2;
    end
end
RMSE_PCA=sqrt(RMSE)

% Correlation Coefficients

for i = 1:3
    for j = 1:3
        correlation_coeff(i,j) = corr2(PureComp(i,:),PureCompPred(j,:));
    end
end

correlation_coeff_PCA=correlation_coeff

figure(1)
plot(300:2:650,PureComp(1,:),'b-')
title('Pure Co Spectrum')

figure(2)
plot(300:2:650,PureCompPred(1,:),'g-')
title('Predicted Pure Co Spectrum')

figure(3)
plot(300:2:650,PureComp(2,:),'b-')
title('Pure Cr spectrum')

figure(4)
plot(300:2:650,PureCompPred(2,:),'g-')
```

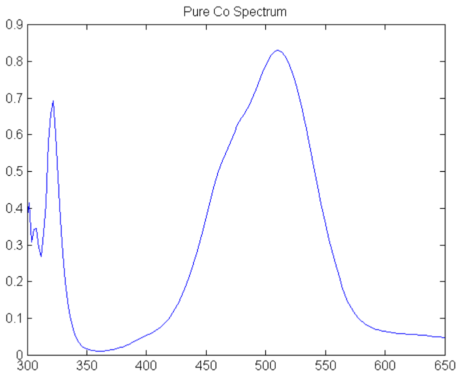
```
title('Predicted Pure Cr spectrum')

figure(5)
plot(300:2:650,PureComp(3,:),'b-')
title('Pure Ni spectrum')

figure(6)
plot(300:2:650,PureCompPred(3,:),'g-')
title('Predicted Pure Ni spectrum')
```

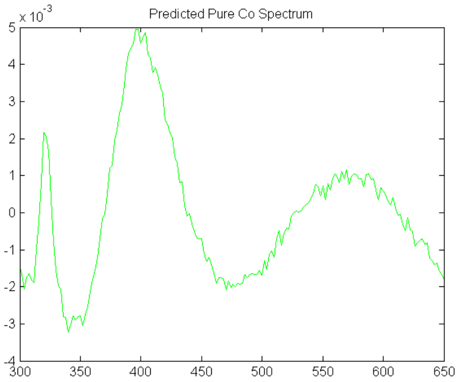
```
RotMat =

     1     1     0
     1     0     1
     0     1     1
     1     0     1
     1     1     0
     1     0     1
     0     1     1
```

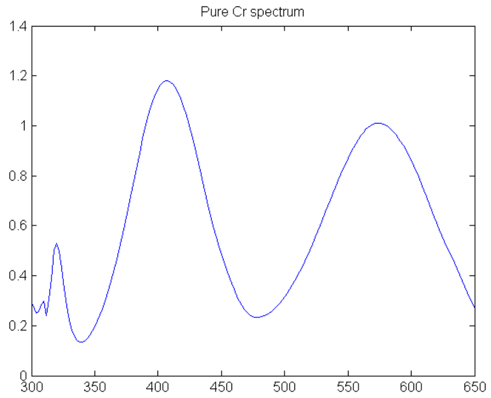
```
RMSE_PCA =

   11.4468
```

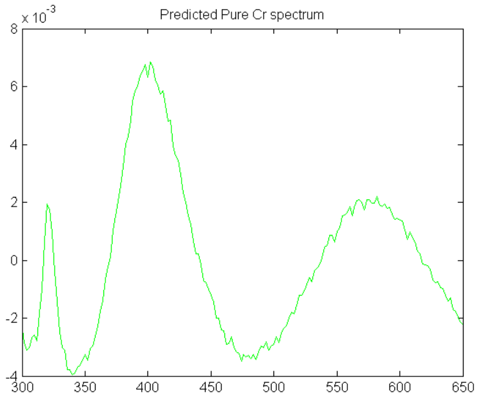
```
correlation_coeff_PCA =

   -0.3102   -0.4490   -0.3993
    0.8930    0.9334    0.6150
    0.7095    0.6857    0.9540
```

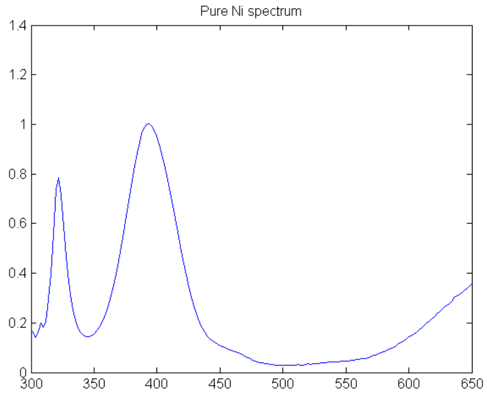
```
* Rows are actual pure components and columns are predicted/extracted pure components
```

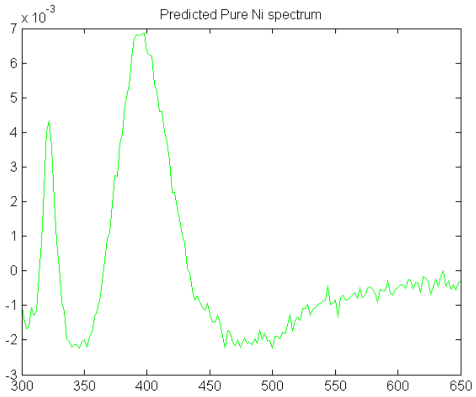












b) NCA

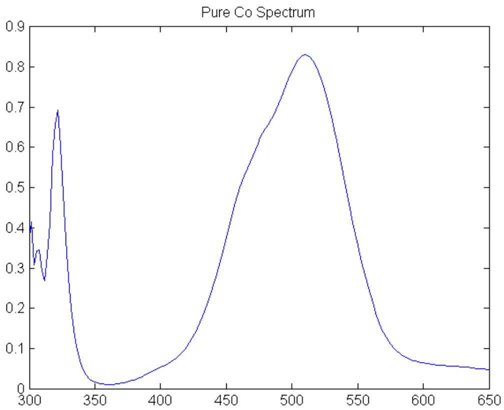
```
% Formatting and saving for NCA solver by Prof. James Liao

Name=[1 , 2, 3];
Number=0:7;
ConMat=cat(1,Name,ConnectivityMatrix);
ConMat=cat(2,Number',ConMat);
save('Connectivity.txt','ConMat','-ascii','-tabs');

Name=1:176;
Number=0:7;
ModelTab=cat(1,Name,ModelData);
ModelTab=cat(2,Number',ModelTab);
save('Model.txt','ModelTab','-ascii','-tabs');

% Applying NCA; running NCA and saving the A and E matrices
Amatrix=load('A.mat');
Ematrix=load('E.mat');

A=Amatrix.A0;
E=Ematrix.E0;
P=pinv(A)*E;

PureCompPred=P;
% Calculating RMSE
```

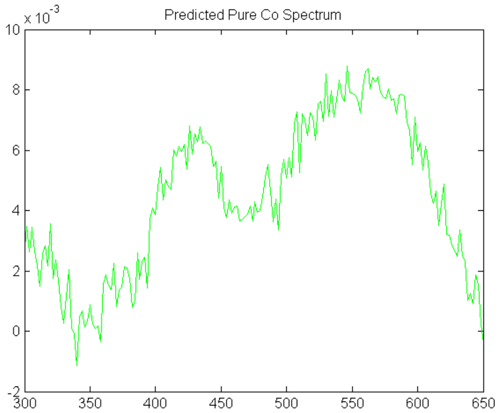
```
RMSE=0;
for i=1:3
    for j=1:176
RMSE=RMSE+(PureComp(i,j)-PureCompPred(i,j))^2;
    end
end
RMSE_NCA=sqrt(RMSE)

% Correlation Coefficients

for i = 1:3
    for j = 1:3
        correlation_coeff(i,j) = corr2(PureComp(i,:),PureCompPred(j,:));
    end
end

correlation_coeff_NCA=correlation_coeff
figure(1)
plot(300:2:650,PureComp(1,:),'b-')
title('Pure Co Spectrum')

figure(2)
plot(300:2:650,PureCompPred(1,:),'g-')
title('Predicted Pure Co Spectrum')

figure(3)
plot(300:2:650,PureComp(2,:),'b-')
title('Pure Cr spectrum')

figure(4)
plot(300:2:650,PureCompPred(2,:),'g-')
title('Predicted Pure Cr spectrum')

figure(5)
plot(300:2:650,PureComp(3,:),'b-')
title('Pure Ni spectrum')

figure(6)
plot(300:2:650,PureCompPred(3,:),'g-')
title('Predicted Pure Ni spectrum')
```

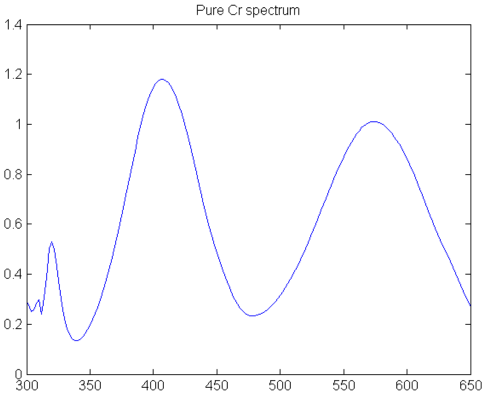
```
RMSE_NCA =

   11.2393
```

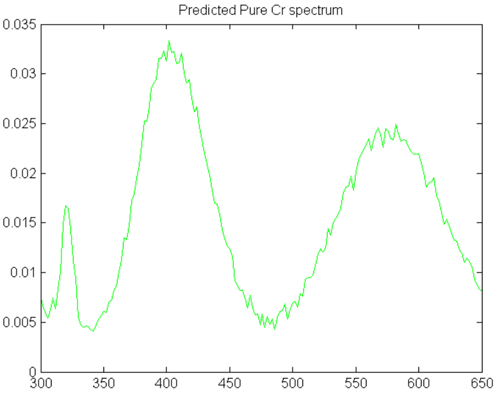
```
correlation_coeff_NCA =

    0.3309   -0.4701   -0.2708
    0.5681    0.9799   -0.0509
   -0.4226    0.5479    0.8944
```

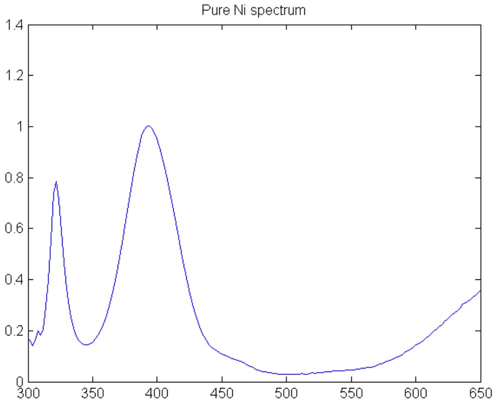
```
* Rows are actual pure components and columns are predicted/extracted pure components
```

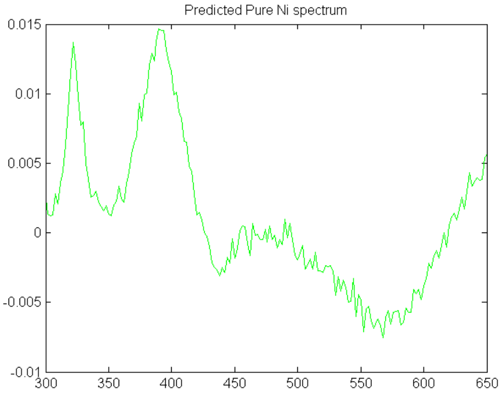












Published with MATLAB® R2014a

PCA Function used in Question 1 and 2:

```
function DenoisedModelData = PCA(ModelData,k)
[M,N] = size(ModelData);
% subtract off the mean for each dimension
mn = mean(ModelData,2);
```

```
ModelData = ModelData - repmat(mn,1,N);
% construct the matrix Y
Y = ModelData' / sqrt(N-1);
% SVD does it all
[u,S,PC] = svd(Y,'econ');
% % calculate the variances
% S = diag(S);
% V = S .* S;
% % project the original data
% signals = PC' * data;
DenoisedModelData=u(:,1:k)*S(1:k,1:k)*PC(:,1:k)';
end
```

Published with MATLAB® R2014a