Name: ANUPARNA BANERJEE
UT EID: AB59958
Email: anuparna.banerjee@utexas.edu

Aim: The aim of this project is to perform *Independent Component Analysis (ICA)* on 5 mixed audio signals to solve the *Blind Source Separation* problem. The results of the algorithm produce 5 decomposed independent signals.

Dataset: There are two different datasets on which the algorithm works on. Both the datasets are available in the *input* folder.
I. *icaTest.mat* contains 3 signals with 40 features each. By mixing them with the given A (*3X3 square* matrix), we created a mixed signal to work on. The produced mixed signals are then used by the algorithm to decompose into independent components.
Ii. *sounds.mat* contains 4s long audio signals. The idea is to mix them and then use the algorithm to retrieve the individual component audio signals.

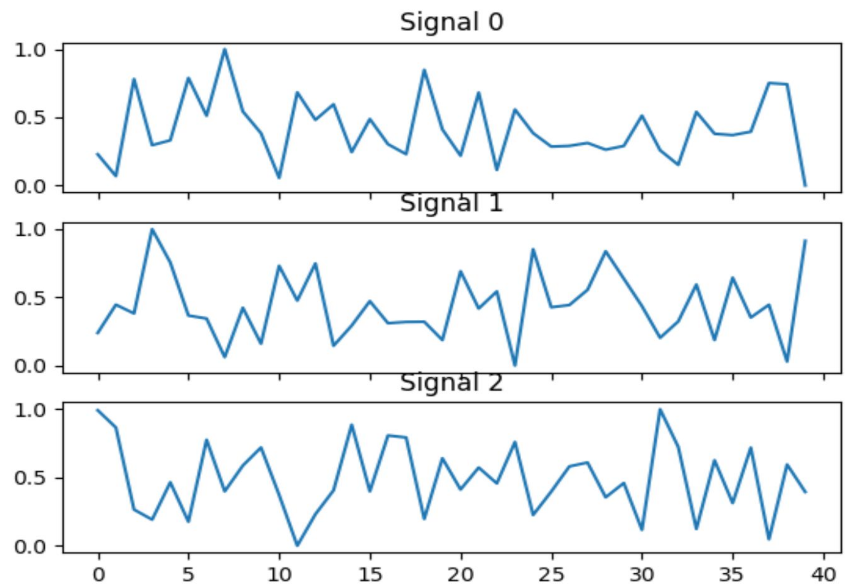
General Description of the Algorithm:

1. Assume $\mathbf{X = AU}$.
2. Initialize the ($\mathbf{n}$ by $\mathbf{m}$) matrix $\mathbf{W}$ with small random values.
3. Calculate $\mathbf{Y = WX}$.
4. $\mathbf{Y}$ is our current estimate of the source signals.
5. Calculate $\mathbf{Z}$ where $\mathbf{z_{i,j} = \mathit{g}(y_{i,j}) = 1/(1+e^{-y}{}_{i,j})}$ for $\mathbf{i} \in \mathbf{[1..n]}$ and $\mathbf{j} \in \mathbf{[1..t]}$ (where $\mathbf{t}$ is the length of the signals).
6. This helps us traverse the gradient of maximum information separation.
7. Find $\mathbf{\Delta W = \eta(I + (1\text{-}2Z)Y^T)W}$ where $\boldsymbol{\eta}$ is a small learning rate.
8. Update $\mathbf{W = W + \Delta W}$ and repeat from step 3 until convergence or R_max iterations (you get bored and decide it is done).

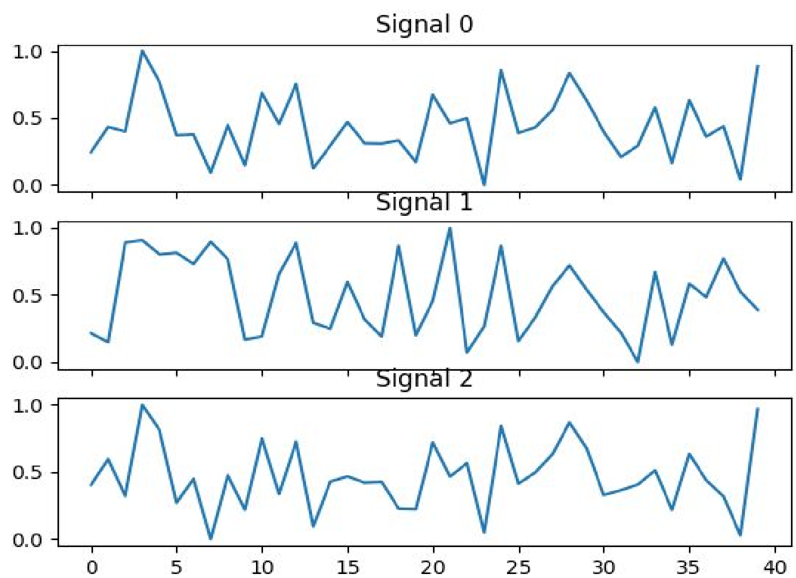
Experiment:

- *Test-data (icaTest.mat) experiment:*
  Using the following configuration on the test-data (icaTest.mat), the plots that follow represents how closely the recovered signals match the source(original) signals.
  Weight matrix, W initial values = Uniform(0, 0.1)
  Step-size = 0.01
  No. of iterations = 1000000

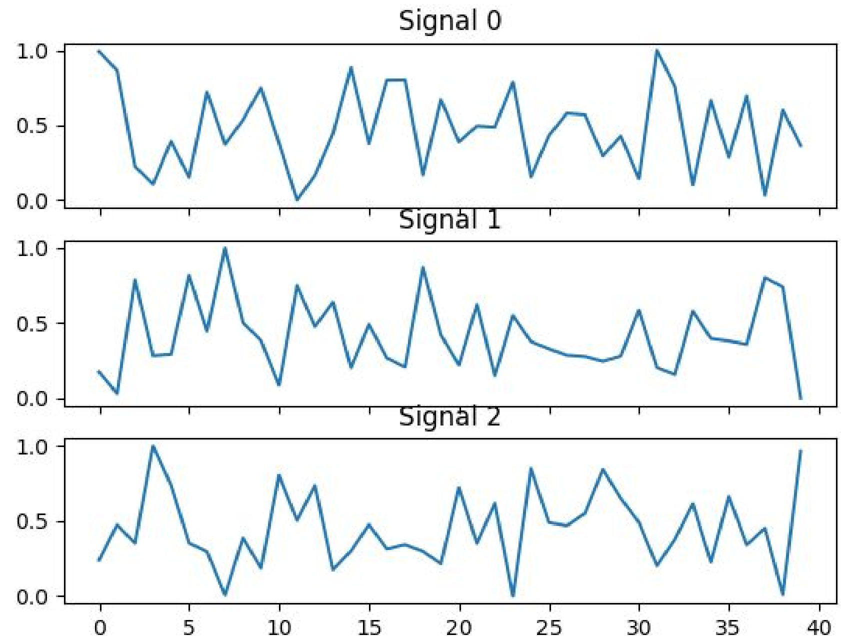
Original Signal Plot



Mixed Signal Plot



Recovered Signal Plot

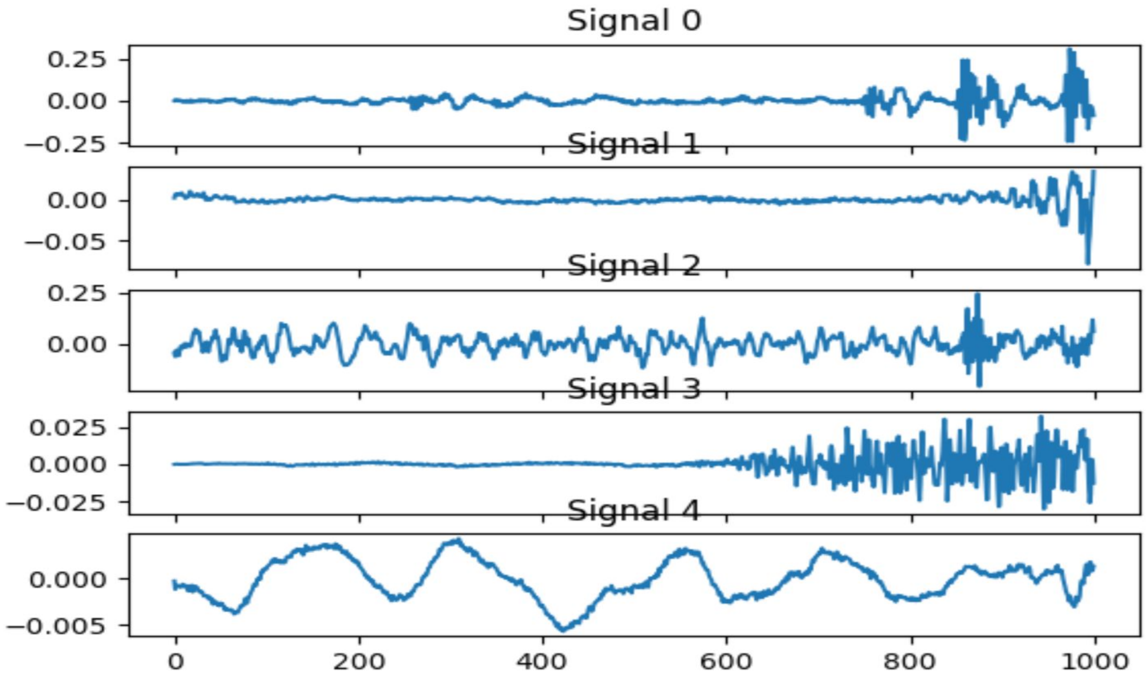


We find that the recovered test-signals are quite similar to the original source signals although in a different order.

- *Actual-data (sounds.mat) experiment:*
  Using the following configuration on *sounds* variable embedded in *sounds.mat*, the plots that follow represents how closely the recovered signals match the source(original) signals.

  The total number of features available in this dataset is 44000 with 5 different signals. To produce a legible plot, the below plot shows the 1st 1000 features of the signals.

Original Signal Plot



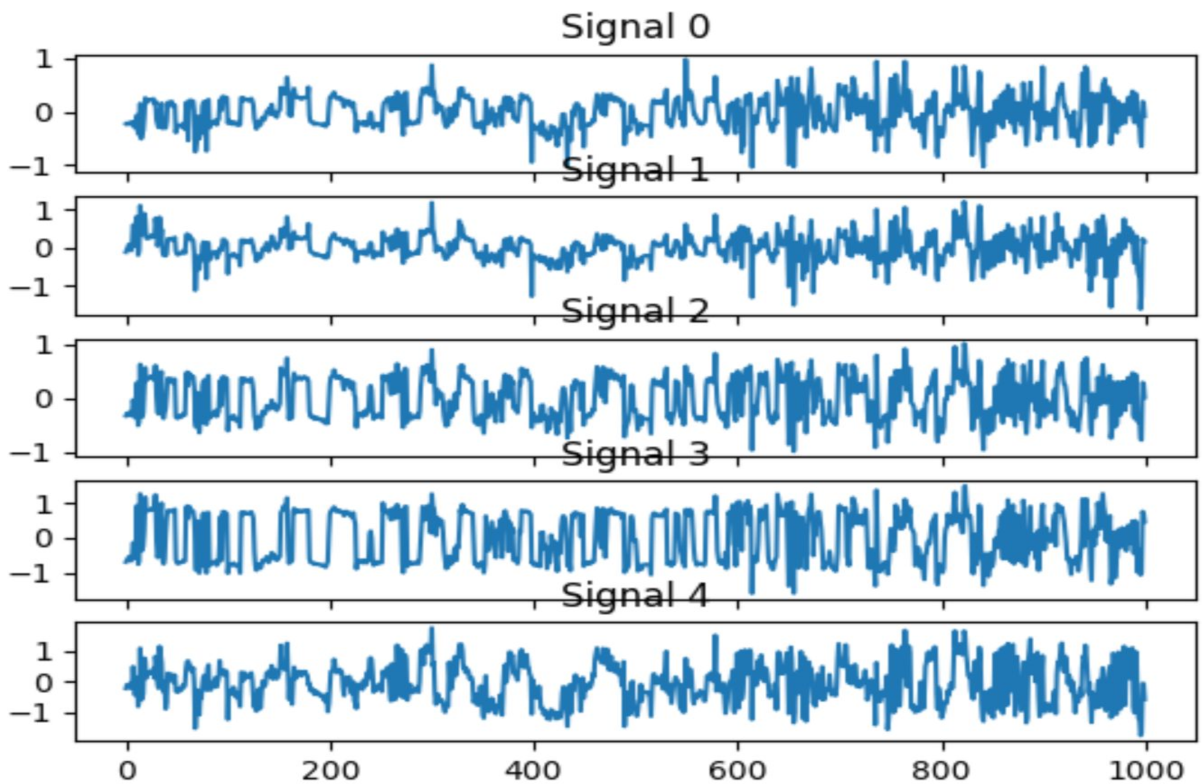
The audio signals corresponding to the source signals can be found as *.wav* files in the folder *\audio\original*

The original signals were then mixed with a 5X5 matrix of random numbers.

The corresponding plot (mixed signals) generated for the first 1000 features (to make a legible plot) is as follows. All plots are available in the folder *figures.* Plots of the original and the mixed signals of the first 10000 features are in the folder with file-names *figures/original-data-signals10000.png* and *figures/mixed-data-signals10000.png* respectively.

The corresponding audio signals of the mixed signals can be found in the folder *\audio\mixed*

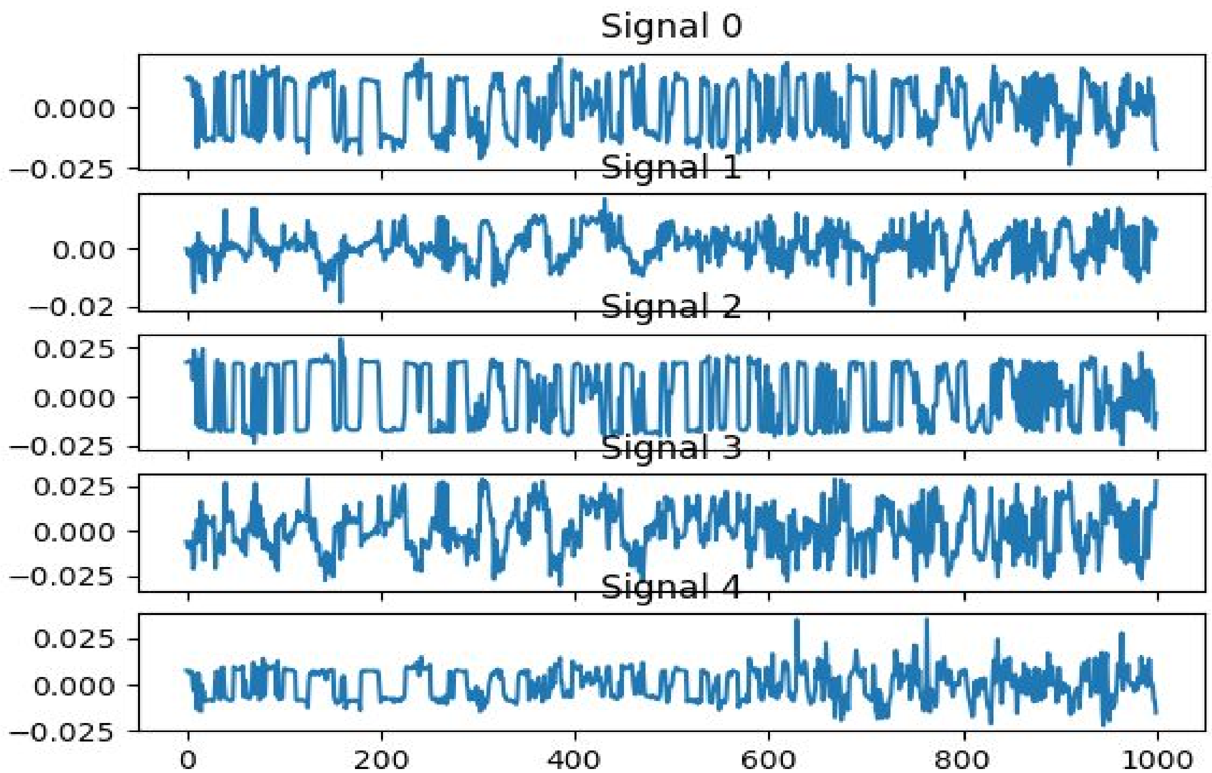
Mixed Signal Plot



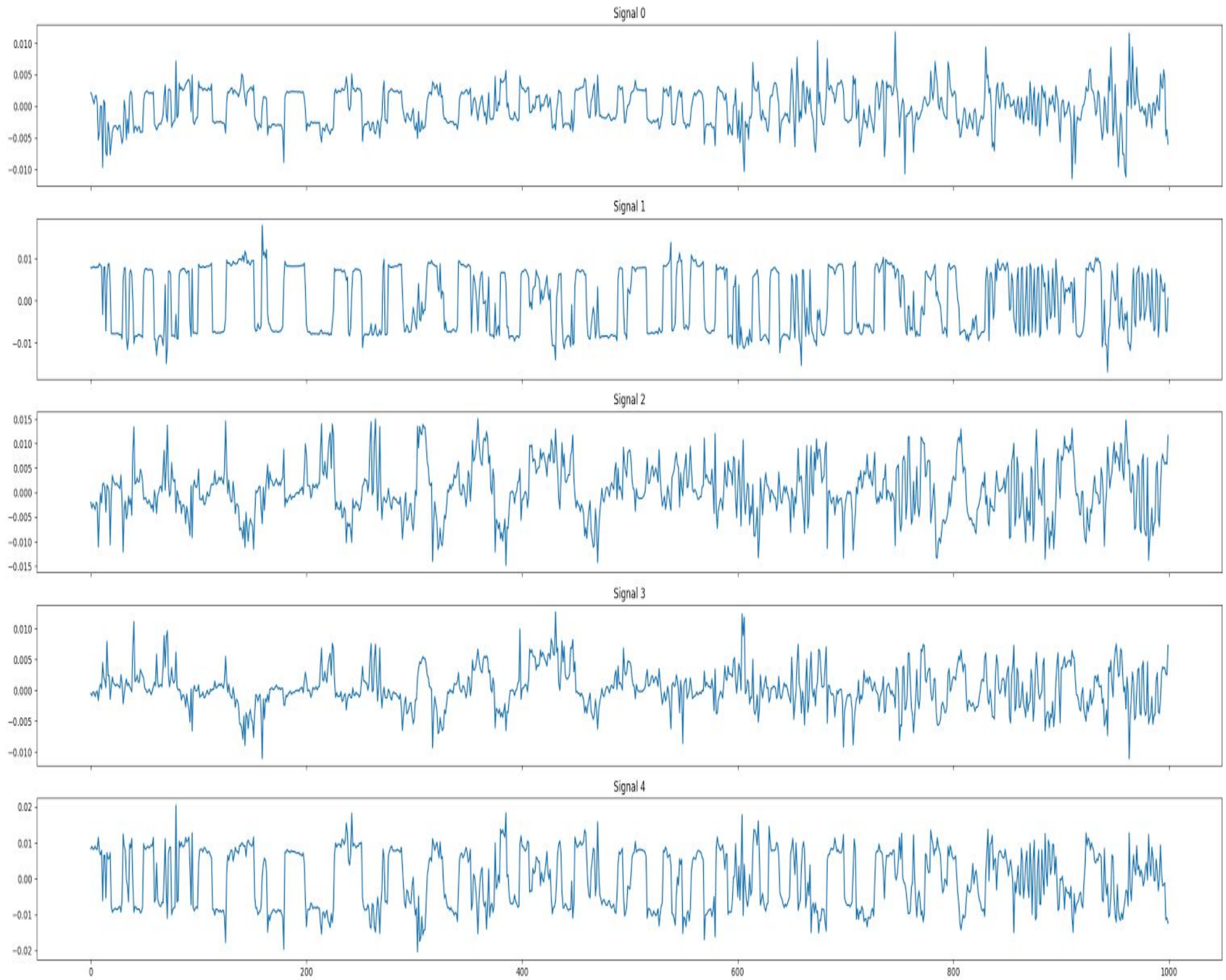
Signal Recovery

Different configurations were tried to recover the original signals.
One of the important points to consider here is the recovered signals might not maintain the same order as the original source signal.
Plots along with the configurations are available below:

- Weight matrix, W initial values = Uniform(-0.01, -0.1)
  Step-size = 0.0001
  No. of iterations = **1000**



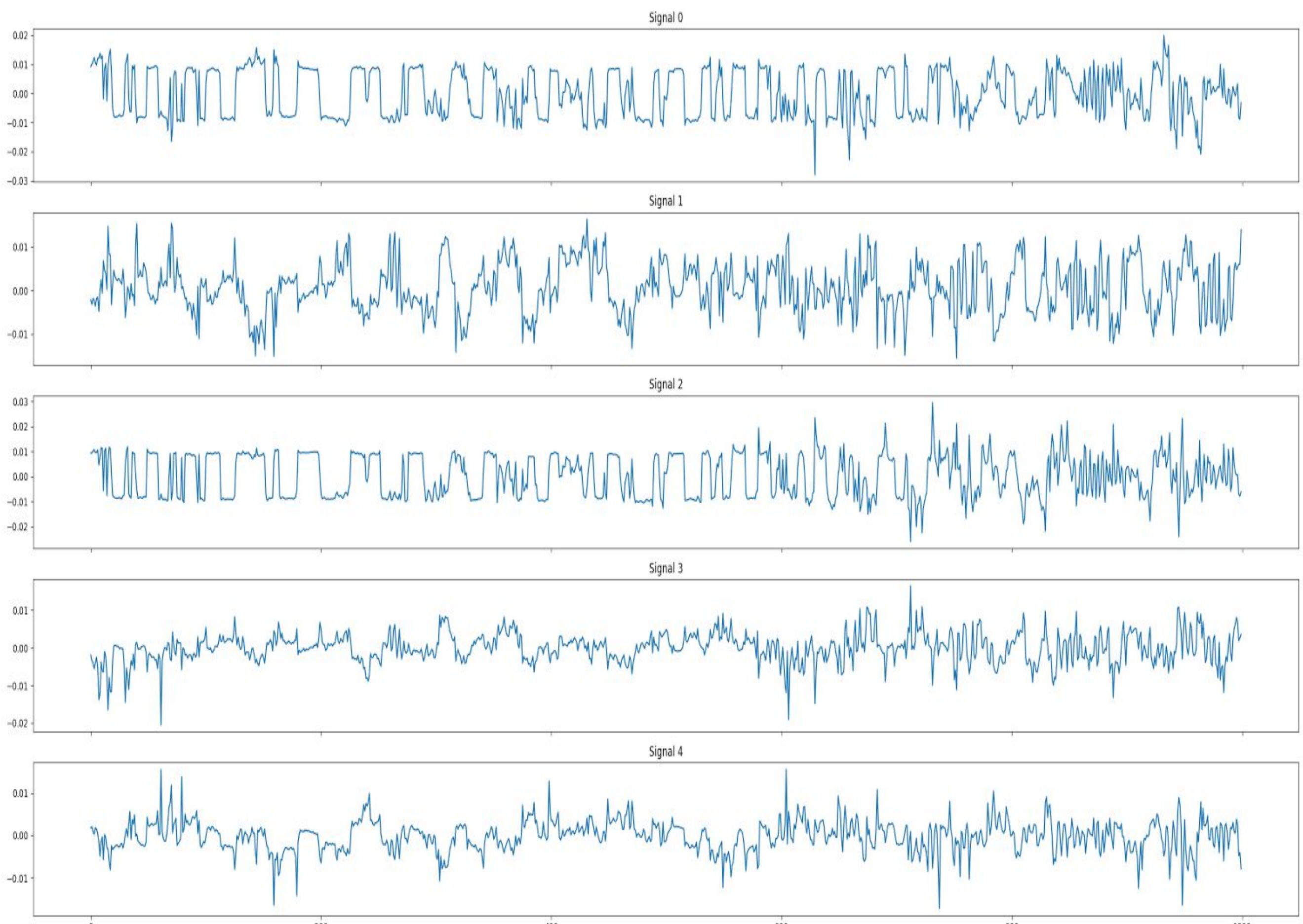
- Weight matrix, W initial values = Uniform(-0.01, -0.1)
  Step-size = 0.0001
  No. of iterations = **10000**



- Weight matrix, W initial values = Uniform(-0.01, -0.1)
  Step-size = 0.0001
  No. of iterations = **100000**

  The recovered *audio files* for this test is present in the folder *\audio\recovered\Iterations_100000_with_step_size_0.0001*

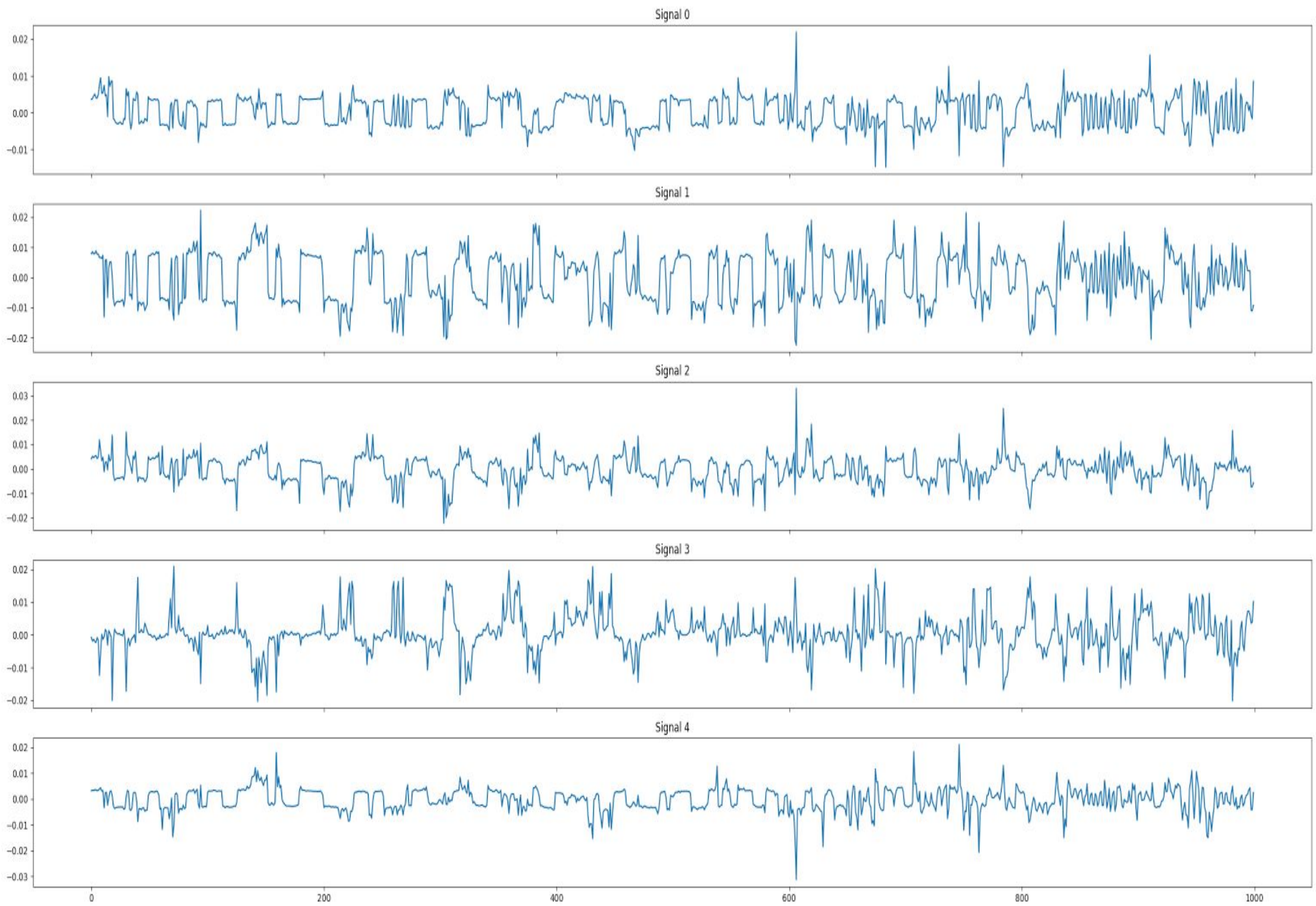
  The recovered *audio files* for this test is present in the folder *\audio\recovered\Iterations_100000_with_step_size_0.0001*



- Weight matrix, W initial values = Uniform(-0.01, -0.1)
  Step-size = **0.0003**
  No. of iterations = **100000**

  The recovered *audio files* for this test is present in the folder *\audio\recovered\Iterations_100000_with_step_size_0.0003*

  The recovered *plot file* for this test is present in the folder *\figures\Iterations_100000_with_step_size_0.0003*



Remarks:

It seems that two (one with a *Homer Simpson Thermodynamics quote* and the other with a *loud laugh of someone*) of the 5 audio signals could be recovered to a certain scale. It would be interesting to note the approximate factor (mathematical form) of the quality of signal recovered to the original signal. Also, another experiment that could be tried is to use only these two signals and doing the same activity (as described above) on them.

Also, if the step size is not very small ~0.0001, the result produces Not a Number errors during Matrix computation.

References:

- https://stackoverflow.com/questions/10357992/how-to-generate-audio-from-a-numpy-array
- https://stackoverflow.com/questions/1735025/how-to-normalize-a-numpy-array-to-within-a-certain-range
- A. Ng. Independent Component Analysis