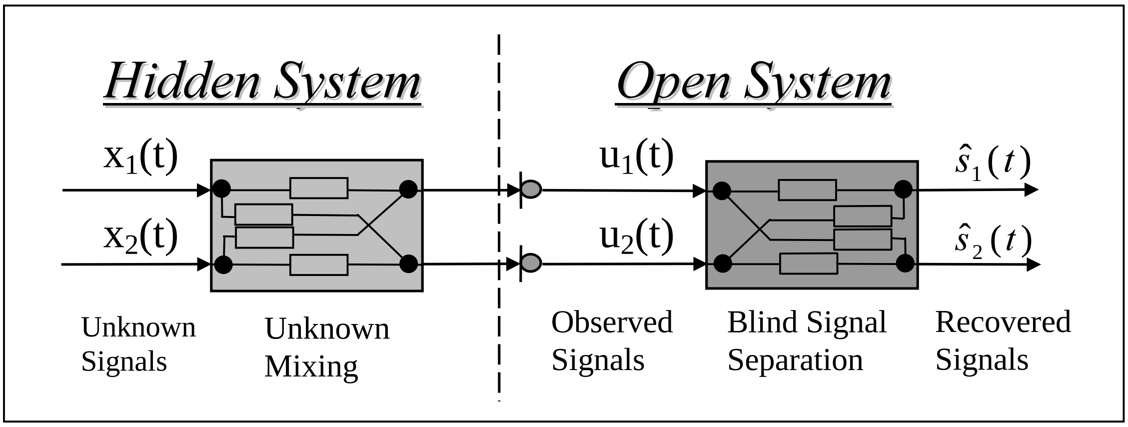
Programming task 4

Blind Signal Separation
with
Independent Component Analysis (ICA)
by means of Neural Training

Independent Component Analysis

In this task we want to recover original sources from observed mixtures. The sources are mixed by a scalar mixing matrix and from the observations we want to find the independent components in order to recover the original sources. The figure below shows a schematic illustration of the problem with 2 sources/mixtures, where the hidden system is unknown, and the aim is to construct the separation system.

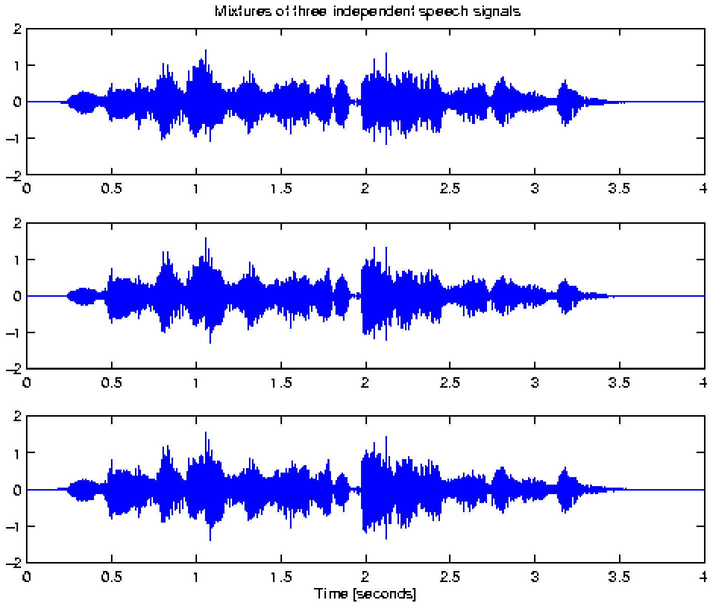


a) In this first part we will create two independent zero mean signals with Laplacian distributions. We then create a mixture of these signals with an arbitrary mixing matrix. From these mixtures we train a Neural Network, by using the ICA-algorithm, and monitor the result.

- Create two sequences (realizations) with Laplacian distributions. The length of the sequences should be 10000, the mean value should equal zero and the variance should equal 2.
  You can use the attached m-file: lapdist.m, to create these signals. Type help lapdist to see the usage of the input parameters. The parameter, b, controls the variance of the stochastic variable and the relation is: $\text{variance} = 2b^2$.
- Create any mixing matrix, A (of size 2-by-2), with full rank (i.e. the inverse should exist). Use this matrix and create two mixtures of the two Laplacian sequences.
- Use the ICA learning algorithm with the nonlinearity chosen as the bipolar sigmoid function (i.e the tangent hyperbolic function) and find the separating matrix, W (of size 2-by-2) from the mixtures.
- Check the performance of the separation by plotting the condition of the performance matrix P=W*A, during training. The performance matrix, P, should equal a, possibly rescaled and permutated, identity matrix. This means that the condition of the performance matrix should go towards one during learning.
  (In Matlab: cond.m returns the condition of a matrix in the Euclidian sense, which is the ratio between the largest and the smallest singular value.)
- Try some different choices of the mixing matrix, A, and find the separation matrix, W. Calculate the performance matrix after convergence and present the results.

b) In this part we wish to separate a mixture of three speech signals. The speech signals are attached in the file: mixtures.mat (for version 4 of matlab). The signals are ordered as row vectors in a matrix, U. The duration is 4 seconds and the sample rate is 12 kHz. The figure below show a plot of the three mixtures.

- Listen to the mixtures and write down what you think is said by each person.
- Perform a separation of the mixtures and create three separated signals. Use the same non-linearity as in the first part. (If the separation is not satisfactory after one iteration of the sequence, iterate several times). Can you distinguish what is said by each person?
- Make a plot similar to the one below with the separated signals at each entry.
- Present the final weight matrix, W.


Mixtures of three independent speech signals
2
1
0
−1
−2
0
0.5
1
1.5
2
2.5
3
3.5
4
Time [seconds]