SELECTED TOPICS IN BIOMEDICAL SIGNAL PROCESSING: TENSOR METHODS AND BLIND SOURCE SEPARATION

1. MLPCA

In this section we are given a 9x100x100 tensor that must be analyzed. To do so, I plotted the 3 first dimensions of the 3 different modes of the tensor and tried to see if there is any obvious principal direction. We can observe that both the mode-1 and the mode-2 data point can lay down into one concrete dimension. Apparently, the tensor mode-3 vectors do not have any notorious one, but we will see numerically how we can still throw away some singular values. We must take into account that we can only plot 3-dimensional data, so our analysis must be primarily numerical.

The first five singular values for the **mode-1 vectors** are:

72.0471 34.3254 19.2121 8.7441 8.5928

These singular values are ordered by importance, and we can see that the fourth singular value is almost 10 times lower than the first one. Hence, we will only keep the first three singular values. In the following figure I have plotted the original data, in blue, and the truncated data in three dimensions, in red. If we had chosen the number of singular values by only looking at the plot of the first three dimensions, we would have probably picked only 2 singular values. As it is shown in green, this would have worked for the three first dimensions, but we would have lost a lot of information in the last six dimensions.

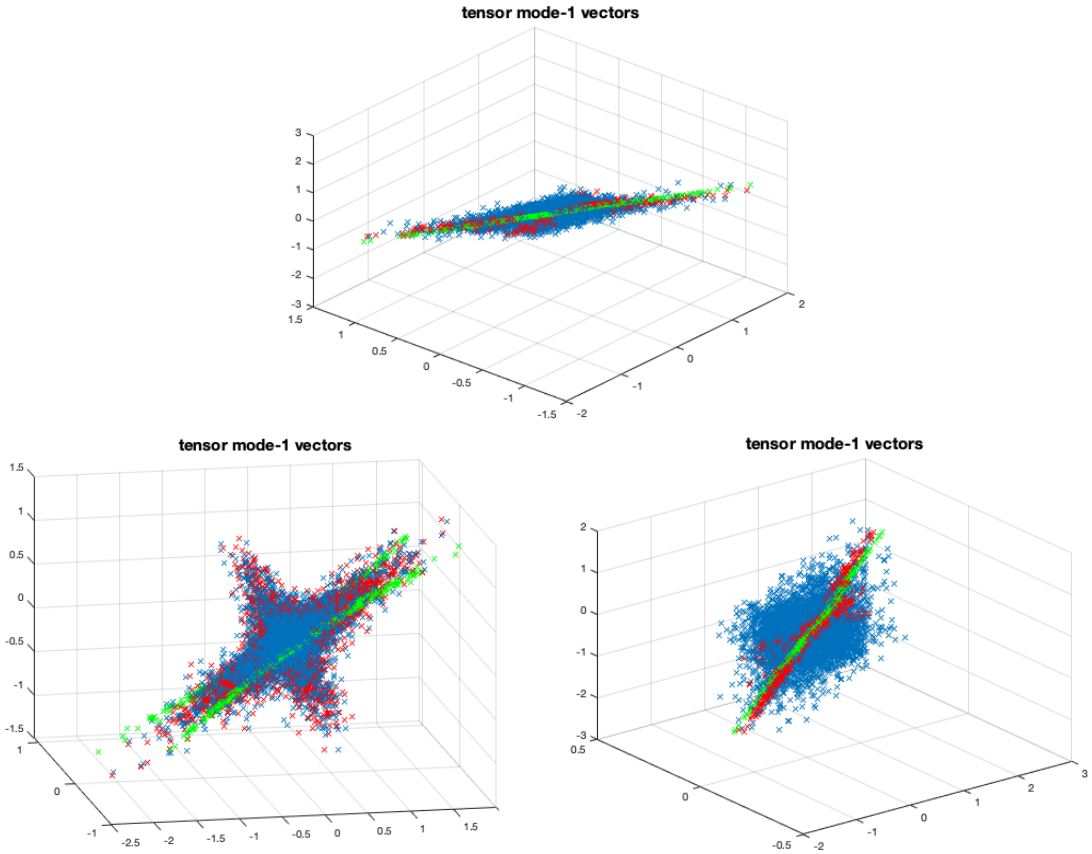


Figure 1: tensor mode-1 vectors

The first five singular values for the **mode-2 vectors** are:

66.1376 41.2278 21.6175 3.3806 3.3311

In a similar way, by looking at the singular values I decided to pick the first three. By comparing the original and the truncated data in the plot we can conclude that it is a good low rank approximation that captures mostly all the information.

The first five singular values for the **mode-3 vectors** are:

66.3030 41.3164 20.9406 3.3244 3.3032

2. Source separation with ICA and PCA

The Blind Source Separation (BSS) can be done by Independent Component Analysis (ICA) or by Principal Component Analysis (PCA); in this section we will explore both methods for source separation. We worked we three different signals, randomly created from a uniform distribution over [-0.25 0.25], and we had different signal-to-noise ratio, SNR, values, from 0 to 50 dB, that would have an effect on the matrix of observation X, in which we would test the quality of the source separation; each of this SNR values was tested 100 times. To sum up, the observation matrix has been obtained by:

$$X = AS + N$$

Our aim is to find the matrix S containing the three sources.

ICA Source Separation

For the ICA source separation, we used the aci.m function. This function receives the observation data Y and returns a matrix F such that $Y = F * Z$, where Z is formed by uncorrelated and independent components. In this case, Z corresponds to our source matrix and Y to the observation matrix X; hence, the resulting matrix would be the estimation of the mixing matrix A.

One measure for the quality of the separation is the signal-to-interference ratio SIR, obtained by the function sir.m. This function receives the two matrices that must be compared, i.e., the true A matrix and the estimated A matrix, and calculate it. It also returns two scaling matrices that has been used in the calculation, but we do not need them in this section.

As mentioned before, we have run the experiment 100 times in which new random signals were calculated, with 11 different SNR ratios. Thus, we obtained a SIR matrix of dimensions 100x11. I calculated the mean for each of the SNR values and represent it in Figure 2.

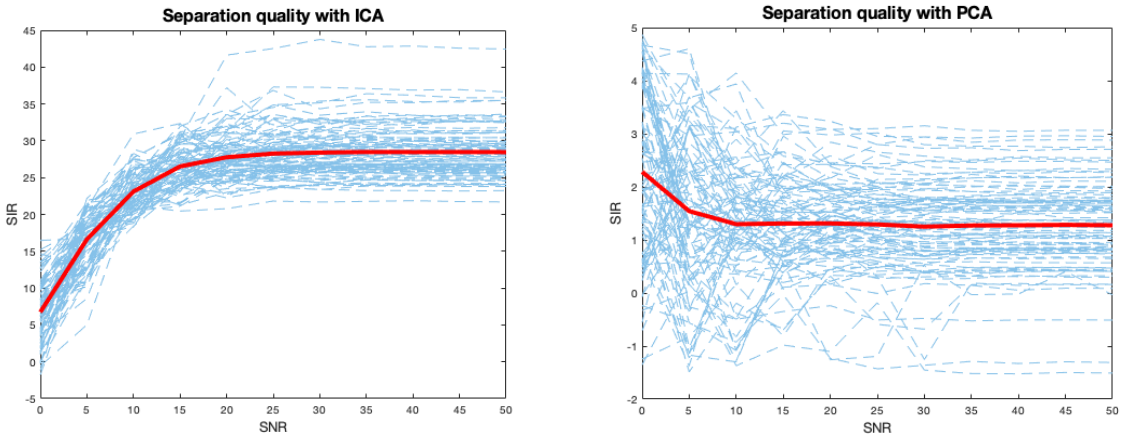


Figure 2: Quality of source separation with ICA and PCA

1.1. PCA Source Separation

While ICA is used to obtain the independent sources that led to the observation matrix, PCA is used to extract the most important features that formed the observation matrix. Despite being different procedures, they are highly related, and they are performed in a similar way.

In this case, we used the MATLAB function pca() that receives the observed data and returns the principal components coefficients. As in the ICA section, we introduced the estimated matrix and the original matrix on the sir function, obtained the SIR values and calculated the mean of the 100 iterations. Again, we can see the resulting qualities on Figure 2.

In the following figure I plotted the estimated sources, calculated with the estimated mixing matrix and the observed data. In light green there are the sources estimated for each of the SNR values, in blue the mean of those sources and in red the real sources that were used to create the observations.

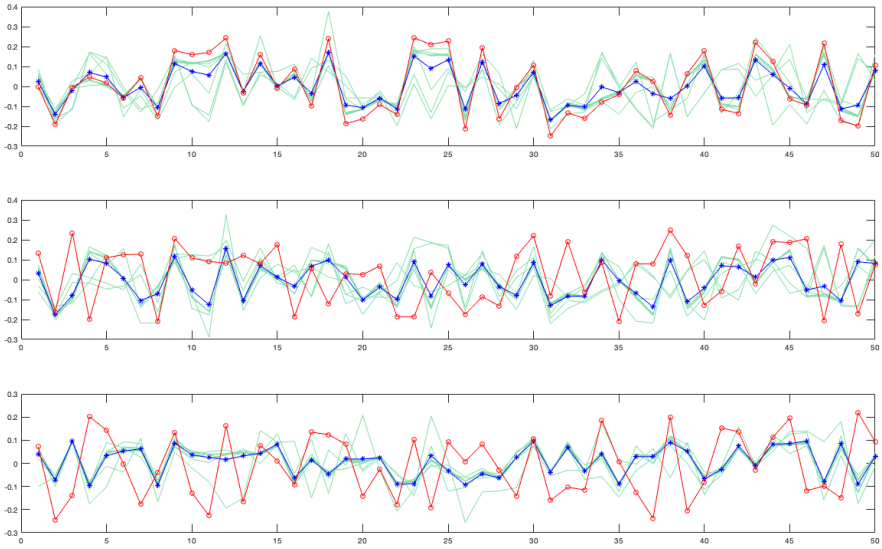


Figure 3: Estimated sources with ICA

3. Synthetic CP

1.2. PCA

In this exercise we are given three matrices, A, B, and C, that have been used to construct the fourth slide of the tensor T by a linear combination. These three matrices are represented in Figure 4. The aim of this exercise is to recover these matrices using PCA.

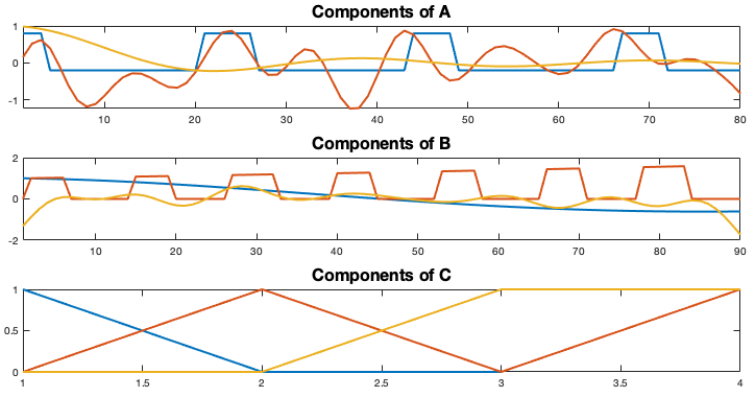


Figure 4: Matrices A, B and C components

We want to decompose the fourth slide, i.e., a data matrix, in rank-1 terms in order to determine the mixing matrix and the source signals. In this case, we can treat our matrix A as the mixing matrix B as the source signals and multiply the result by the factors contained in the fourth row of the C matrix; as we have seen, if we wanted to recover the whole tensor T, the *r-th* diagonal of the core tensor is the *r-th* row of the matrix C; hence, we will take the fourth row.

I first computed the SVD of the fourth slide T4. We can observe that there are three singular values different to 0, meaning that it has three principal sources; this verifies what we knew *a priori*. We would need to calculate three rank-1 terms to recover the matrices A, B and C.

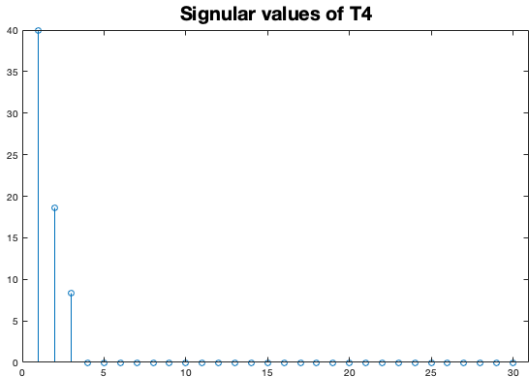


Figure 5: Singular values of the fourth slide of the tensor T

As in the previous section, I calculated the PCA of T4, obtaining the principal components' coefficients; as we know that there are three meaningful singular values, we take the first three columns of the coefficients and conclude that they form the mixing matrix. In a similar way we calculate the sources matrix. Even though we obtained the exact fourth slide, we cannot conclude that the mixing matrix and the sources are the correct ones because the main assumption of the PCA method is that both matrices are orthogonal, which is not strictly necessary. Hence, the method we must use with this aim is the Canonical Polyadic Decomposition (CPD).

Canonical Polyadic Decomposition (CPD)

The CPD computation consists on the following steps:

Compression using the MLSVD to obtain a smaller tensor. After adding noise with an SNR of 15 dB to the original tensor, I computed its MLSVD and got the following singular values for each of the modes.

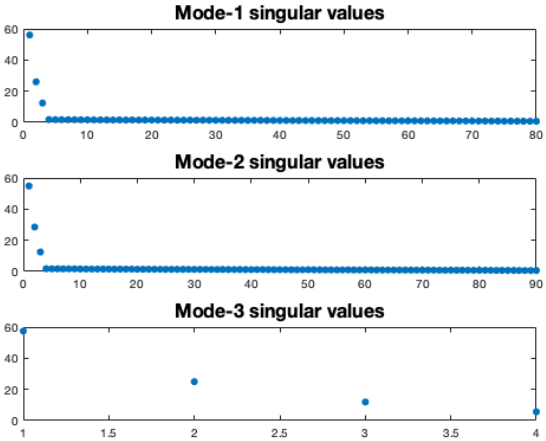


Figure 6: Tensor modes singular values

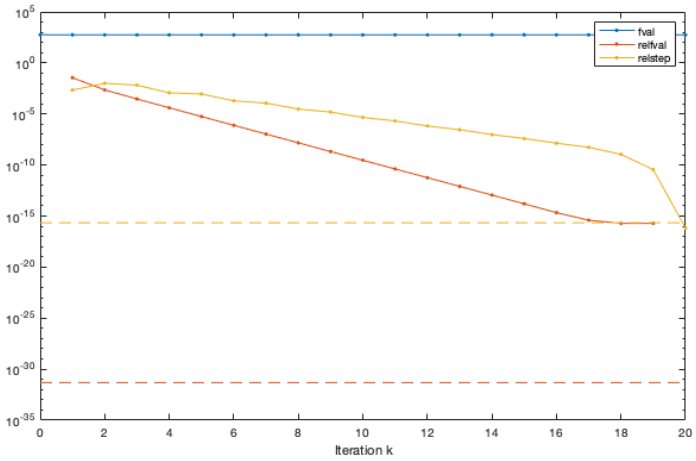
In this exercise I will compute the CPD varying the number of rank-1 terms from 1 to 5. When we plot the singular values for the three modes of the noisy tensor, we can already see that the optimal R will be 3. I also tried with R=4, because in the third mode the fourth singular value could also be chosen as optimal, and R=5 to check that we start to get worse results.

In order to **initialize** the process, we have to choose between the GEVD (Generalized Eigenvalue Decomposition) and a random initialization. Next, we have to choose a **core algorithm for optimization**, which can be the ALS (Alternating Least Squares) or the NLS (Nonlinear Least Squares).

For comparing the results for all the 5 ranks, I used GEVD for initialization and NLS as the core algorithm.

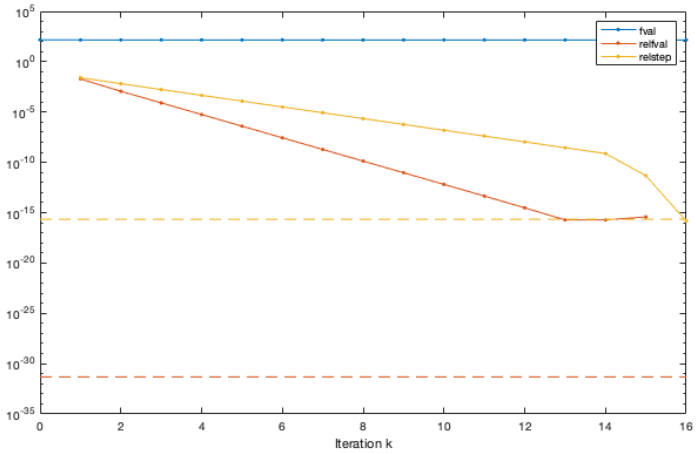
- **R = 1**

In this plot we can observe how the step size tolerance, the dashed yellow line, is reached with 20 iterations. The final relative error is 0.522241.



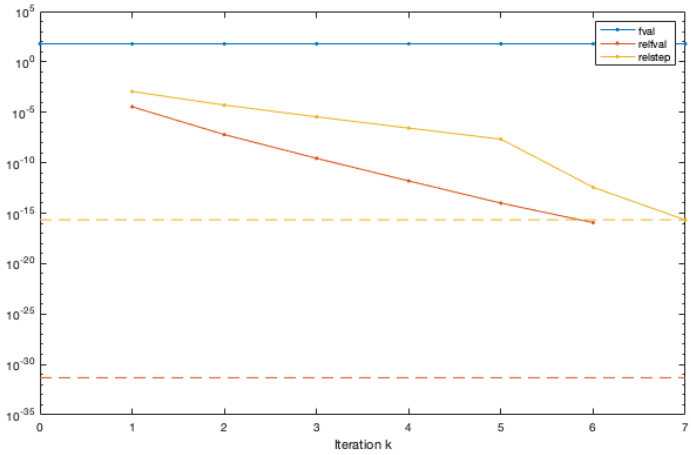
- **R = 2**

In this case, with the rank set to 2, the step size is reached with 16 iterations and having lowered the relative error to 0.266521.



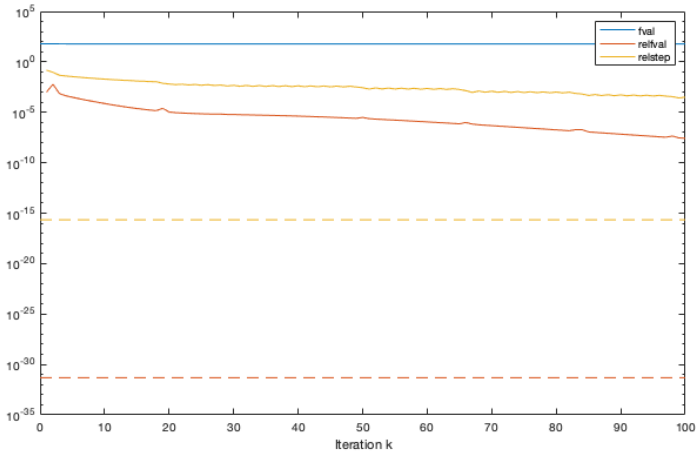
- **R = 3**

When the rank is 3, there are only 7 iterations needed and it reached a relative error of 0.173257. Again, the procedure stopped when the limit step size was reached.



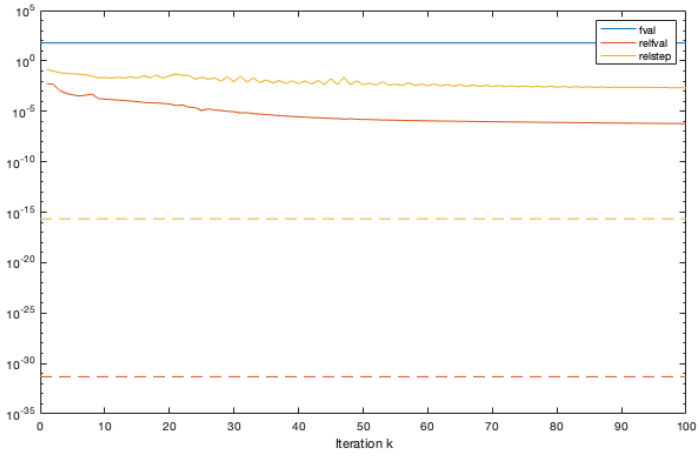
- **R = 4**

In the case of rank 4, the relative error is more or less the same as in the previous case, 0.172209, but it needed much more iterations to obtained it. Moreover, it stopped because it had completed the maximum number of iterations, set to 100.



- **R = 5**

Again, the relative error was similar to the previous ones, 0.171028, but it reached the maximum number of iterations as with rank 4.



In Figure 7 I plotted the number of iterations, the relative errors, and the relation between both magnitudes in order to better see the comparison among all the ranks. As we could predict, rank that gives us the lower relative error and number of iterations is the rank 3.

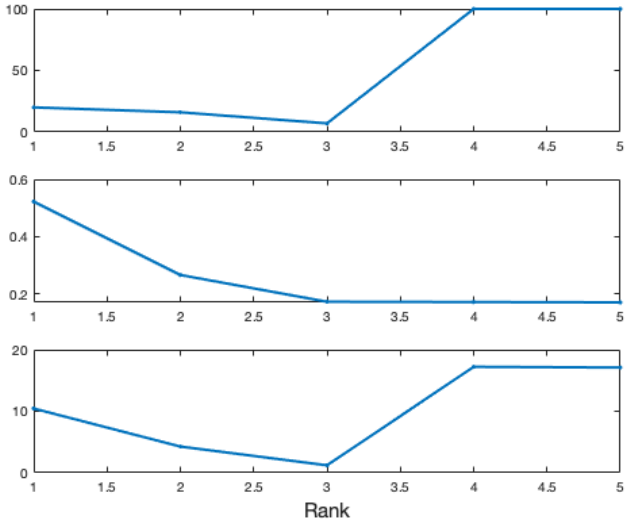


Figure 7: Comparison between the five rank posibilities

Taking the rank 3, the final matrices A, B and C are:

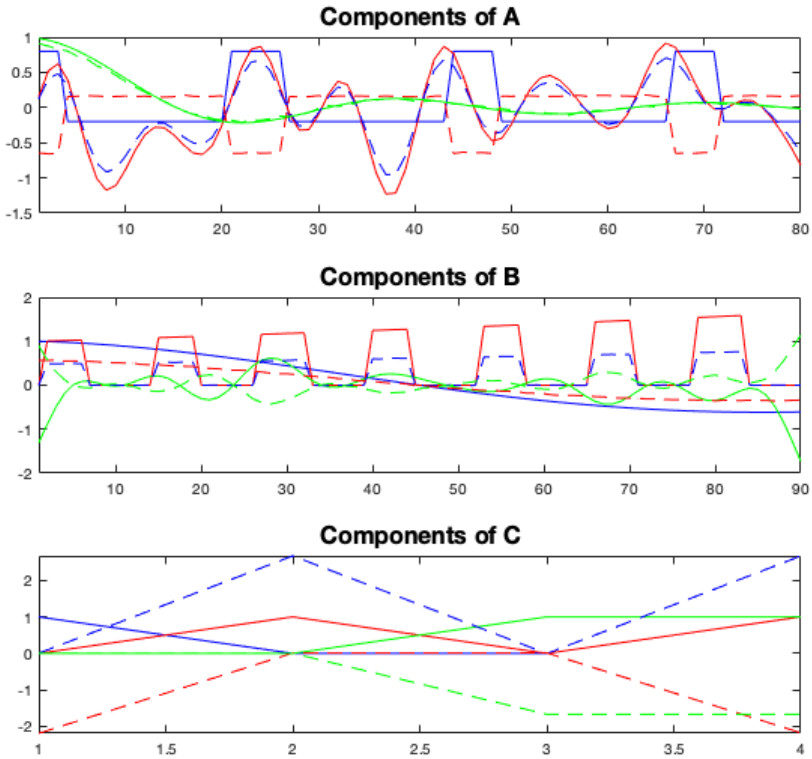


Figure 8: Estimated matrices A, B and C

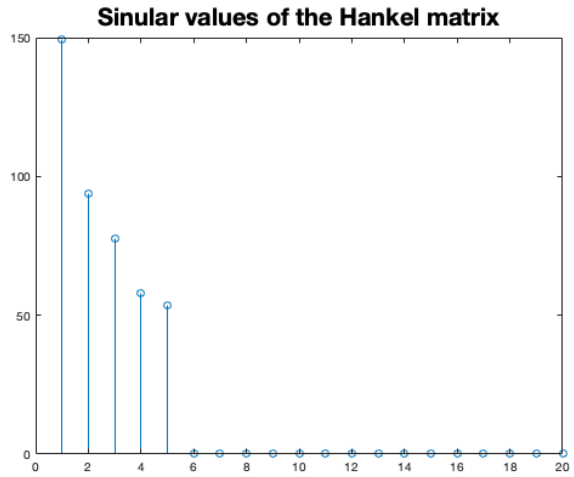
4. Harmonic retrieval

In this exercise we must find the poles of a given system. We will use three different techniques: ESPRIT, Low Multilinear Rank Approximation LMLRA and Canonical Polyadic Decomposition CPD. However, all of them have in common the final calculations. We must find the matrix M

such that $\underline{U} = M\overline{U}$, where $\underline{U}$ is the matrix U omitting the last row and $\overline{U}$, omitting the first row. Once we have this matrix M calculated, we look for its eigenvalues that will be the poles of the system. We must remark that the number of poles of the system is equal to the number of significant eigenvalues. The difference between these techniques is how we obtain the matrix U.

ESPRIT

ESPRIT is a matrix technique. First, we calculate the matrix He by hankelizing the given vector x; as a result, we obtain a matrix of dimensions 300x301. We calculate the SVD decomposition of this matrix and look at the singular values; we can see in XXX how there are only 5 significant ones. Hence, the number of poles is 5. We use the matrix U to obtain $\underline{U}$ and $\overline{U}$ and continue as explained before. We can see the poles obtained by this method in Figure 10 plotted in red.



LMLRA

This is a tensor approach that calculates a compressed tensor of multilinear rank [R1 R2 R3]. We first look for these three values by computing the MLSVD decomposition. Again, we see that in all the three tensor modes the rank is 5, so we will calculate $\mathrm{Ul} = \mathrm{lmlra}(\mathrm{H_{ten}}, [5\ 5\ 5])$. We obtain a structure containing 3 cells; our desired matrix U corresponds to the first cell. Again, we continue as explained before. We can see the poles obtained by this method in Figure 10 plotted in blue.

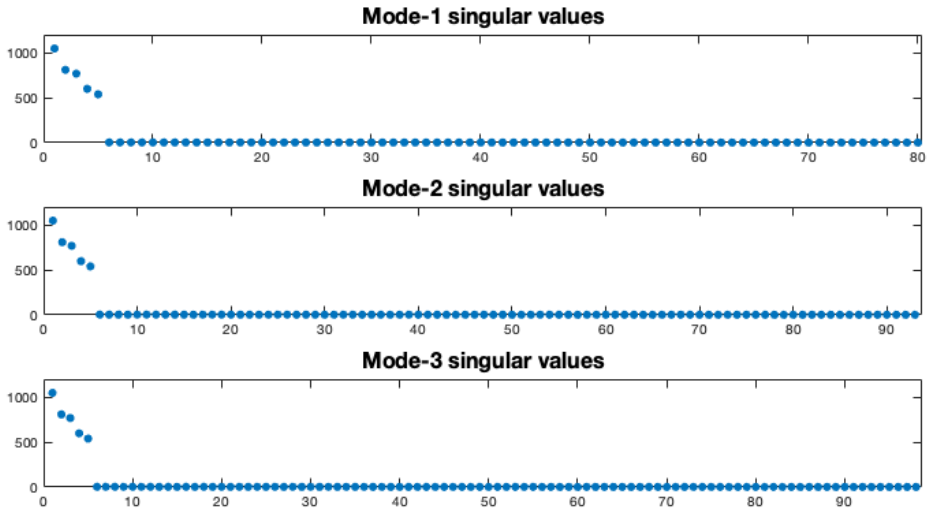


Figure 9: Singular values for the tenor modes

CPD

CPD is another tensor approach. Now, we first need an initialization of the algorithm by @rand. Once we have obtained this tensor, we can run $\mathrm{Uc} = \mathrm{cpd}(\mathrm{H_{ten}}, \mathrm{U_0})$. We obtain another 1x3 cell structure from which we pick the first one to be our U matrix. We can see the poles obtained by this method in Figure 10 plotted in green.

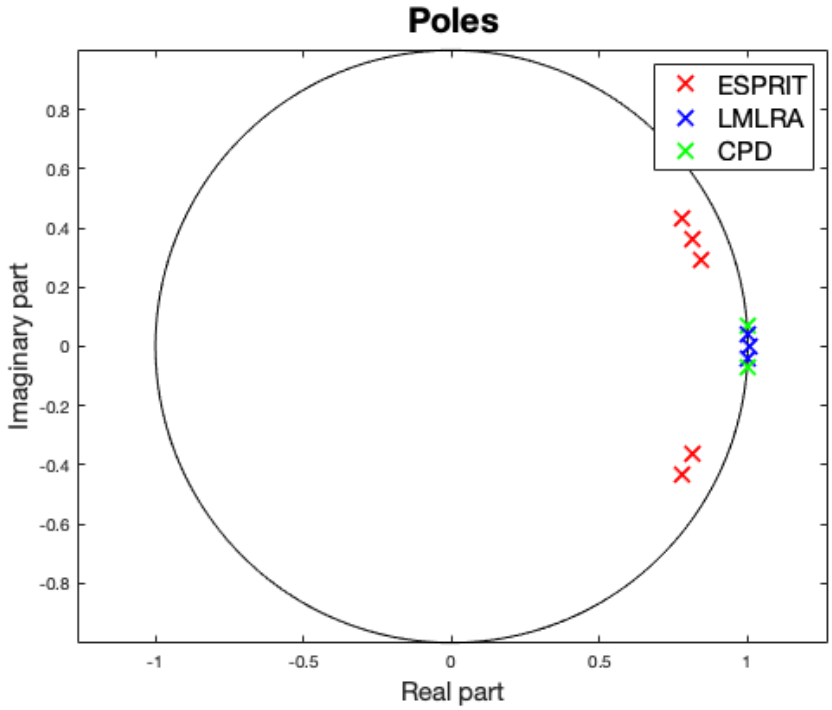


Figure 10: Poles calculated with the three methods