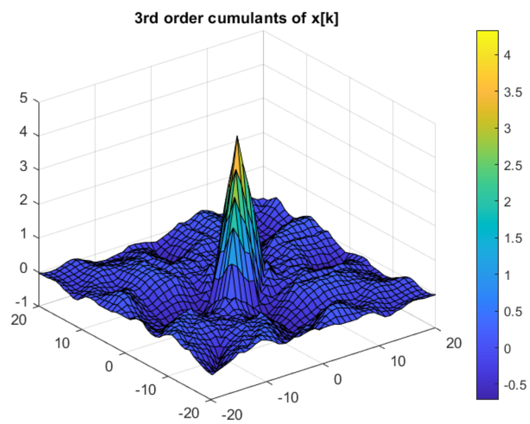
Query 1

Fist, we try to validate the non-Gaussian nature of our input signal v(k). The input signal is generated from a exponential distribution. Thus, we expect that $\gamma_3^v(0,0) \neq 0$. The value calculated from MATLAB was $\gamma_3^v(0,0) = 1.7268$, proving our expectation.

Query 2

Next, we will calculate and plot the 3rd order cumulants of the output signal x(k) from $\tau 1 = \tau 2 =$ (-20:0:20). In order to estimate the 3rd order cumulants, we will use the indirect method via the function Cumulants3.
Hence, we got the following plot.



Query 3

Next, we will apply Giannakis' formula in order to estimate the coefficients of the MA system transfer function via the estimations of the 3rd order cumulants.
Giannakis' formula states:

$$\hat{h}[k] = \begin{cases} \frac{c_3^x(q,k)}{c_3^x(q,0)}, & k = 0, 1, .., q \\ 0, & k > q \end{cases}$$

Thus, we need to know the order q of our MA system. In our case, q = 5.

Query 4

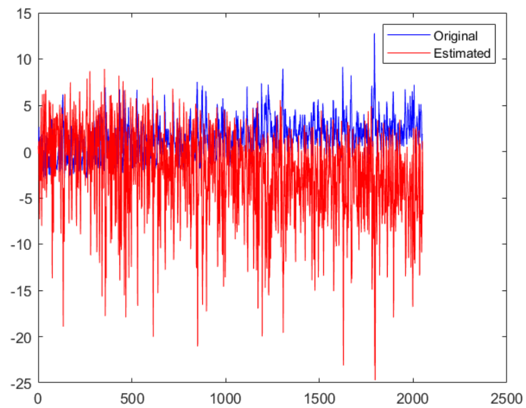
In order for the Giannakis' formula to be applied we must know the order q of the MA system. In some cases, there are deviations on the calculation of the correct order q leading either to sub-estimations or sup-estimations.
In this assignment we will examine two particular cases.

- qsub = q - 2 = 3
- qsup = q + 3 = 8

For each case, we estimated with Giannakis' formula the $\hat{h}sub$ and $\hat{h}sup$.

Query 5

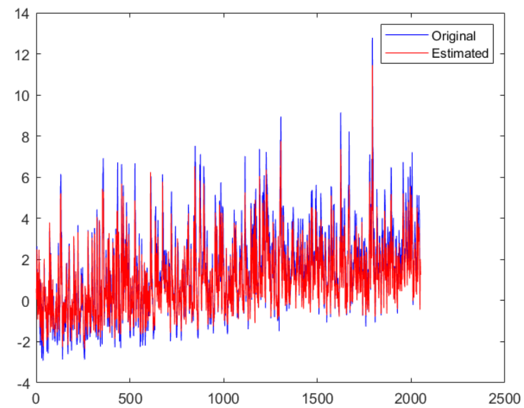
Here, we will estimated the output of the MA system $xest$ using the impulse response $\hat{h}est$ from Giannakis' formula. In order to compare the estimated output $xest$ with the original output x, we calculated the NRMSE between them.
The NRMSE was found to be NRMSE = 0.4479.
The following plot shows the $xest$ vs x.

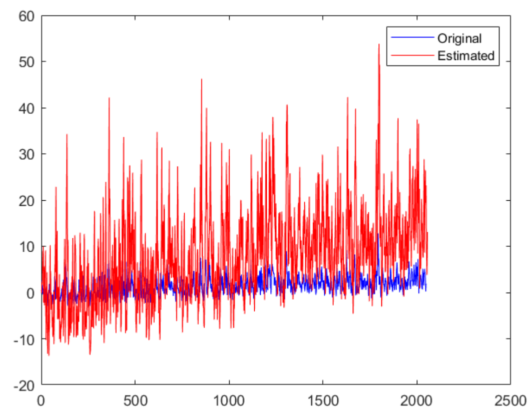


Query 6

In this step, we will repeat query 5 using $\hat{h}sup$ and $\hat{h}sub$ as our impulse responses of our system. Similarly, in order to compare the estimated outputs with the original output x, we calculated the NRMSE between them.
The NRMSE for the sub-estimation was NRMSEsub = 0.0512.
The NRMSE for the sup-estimation was NRMSEsup = 0.7264.
NRMSEsub is way better than NRMSE from the estimation of actual order q. Hence, Giannakis' formula performs better when we pass as a parameter a sub-estimation of the actual order of our MA system. In our case, the formula performs better when the order q = 3 while the actual order is q = 5.
We can assume that this is an indicator of the formula's better impulse response estimation when the MA system has a low order (q<5).
This assumption is also validated by the fact that the NRMSEsup is larger than the NRMSE of the actual order, possibly indicating that the formula not only estimates better with low order MA systems (q<5), but also the estimations seem to be completely wrong when the system has a high order (q>5).

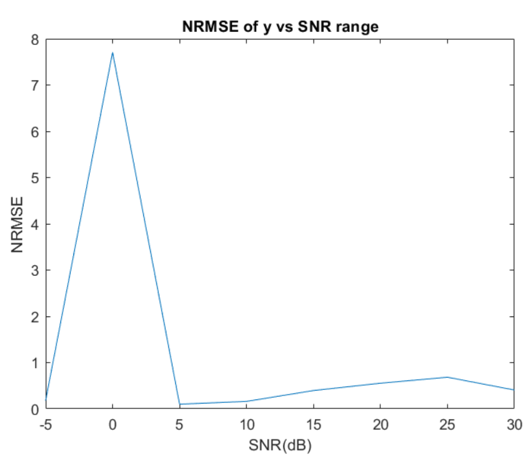
The first plot shows the x sub-estimated vs x.
The second plot shows the x sup-estimated vs x.





Query 7

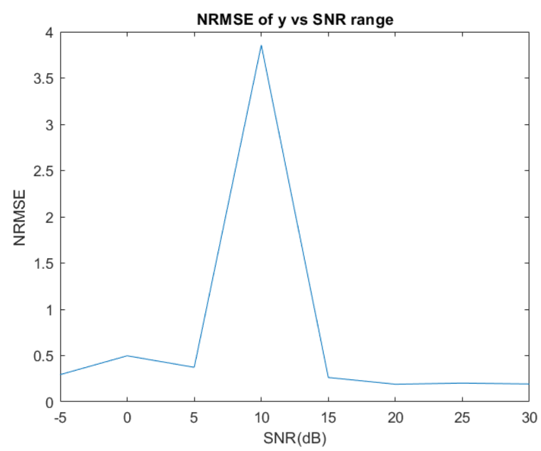
In this step, we examined the performance of Giannakis' formula if we add noise to the signal output. The signal output for this query is y_i[k] = x[k] + n_i[k], i = 1,2,...,8 where n_i is white Gaussian noise such as the SNR will be [30,25,20,15,10,5,0,-5] dB for each i respectively. The NRMSE is computed for each i and plotted against the corresponding SNR values.
Hence, we got the following plot.



We can state that for medium to high SNR values ([5 to 30]) the NRMSE in not affected by the additive noise. This means that Giannakis' formula is robust to the addition of white Gaussian noise when the noise contributes less than the actual signal.
Interestingly, for SNR = 0, when the noise contributes equal with the actual signal, the NRMSE peaks at NRMSE = 7.7064 and for SNR = -5 dB, when the noise contributes more than the actual signal, the NRMSE returns to low levels, NRMSE = 0.1761.

Query 8

For the last step, instead of using one realizations of the input and output data of the MA-q system, we repeated the whole process 50 times and worked with the mean values of the results. Our assumption that the formula work better with low order MA systems (q<5) and estimates completely wrong for higher order MA systems (q>5) is validated.
The mean NRMSE for the actual order is NRMSE = 0.1817.
The mean NRMSE for the sub-estimation is NRMSEsub = 0.0512.
The mean NRMSE for the sup-estimation is NRMSEsup = 1.2066.
Interestingly, the mean SNR vs NRMSE peaks at NRMSE = 3.85 and for SNR = 10 dB and remains beyond 0.5 for the rest SNR values.
Hence, we got the following plot.



Discussion

Giannakis' formula, a formula used to estimate the impulse response of a MA system of order q using the 3rd order cumulants of the system's output, estimated via the indirect method, works well in a noise-free environment when the order q is known a priori.
The mean NRMSE of the original signal vs the signal estimated using the impulse response from the formula is NRMSE = 0.1817.
Another interesting fact is that Giannakis' formula work even better than the original q order as input when we sub-estimate the order q of the MA system, NRMSEsub = 0.0512. Moreover, the results are disappointing when we sup-estimate the order q of the MA system, NRMSEsup = 1.2066.
The latter results support our assumption that the formula works better with low order MA systems.
Finally, when testing the formula in a noisy envinroment with different SNR values the results are pretty absurd. With the addition of white Gaussian noise, we hypothesized that the NRMSE values would be independent of the different SNR values (e.g plot of NRMSE vs SNR would be a straight horizontal line) because we work with the 3rd order cumulants of the output, noise contaminated signal in order to estimate the impulse response of the MA system which means that any Gaussian terms would be zeroed. The results although show that the NRMSE peaks randomly at one specific SNR value (SNR = 10 dB for Query 8).
As we know from (**Cumulants: A powerful tool in signal processing,1987,Giannakis**) the only error introduced in the formula comes from the 3rd order cumulants calculation.
Thus, we can state that this error is responsible for this absurd behavior.

Matlab code

Follow the homework assignments at github.