Dafni Nikolaidou
10546
dafninikol@ece.auth.gr

Homework Set 3
Advanced Signal Processing

May 13, 2024

Question 1

The input signal $v[k]$ is generated from an exponential distribution, thus we expect a signal of non-Gaussian nature. This can be validated through the skewness, for which $\gamma_3^v \neq 0$ should be true. Indeed, the value calculated using MATLAB is $\gamma_3^v = 1.8378$, proving our initial expectation.

Question 2

We will now calculate and plot the 3rd-order cumulants of the process $x[k]$. For the estimation we will use the indirect method with $K = 32$, $M = 64$ and $L_3 = 20$. The function bisp3cum from MATLAB Central File Exchange is used to produce the following results:

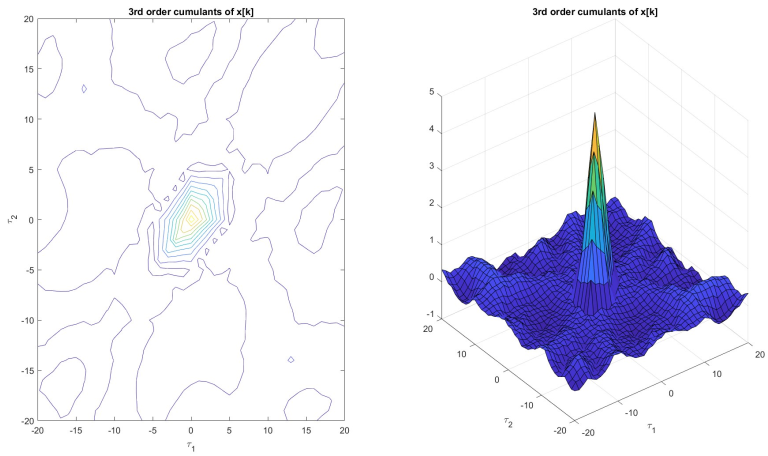

Figure 1: c_3^x estimation using bisp3cum function

Question 3

The estimated impulse response of the MA system using Giannakis formula is:
$\hat{h}[k] = \begin{bmatrix} 1.0000 & 2.1328 & 1.2103 & 1.0356 & 1.1843 & -0.2870 \end{bmatrix}$
Giannakis formula : $\hat{h}[k] = \frac{c_3^x(q,k)}{c_3^x(q,0)}$ is used for order of the system $q = 5$.

Question 4

In this case, considering a divergence from the accurate calculation of the MA system's order q, we set $q_{sub} = q - 2 = 3$ and $q_{sup} = q - 3 = 8$. We estimate the impulse response of the MA system using Giannakis formula, in the same manner as the previous question.

Sub-estimation of the order q

$\hat{h}_{sub}[k] = \begin{bmatrix} 1.0000 & 0.8679 & 0.7829 & 0.6998 \end{bmatrix}$

Sup-estimation of the order q

$\hat{h}_{sup}[k] = \begin{bmatrix} 1.0000 & 2.3845 & 0.9877 & 3.6569 & 1.3345 & 1.2550 & -1.0948 & -3.7393 & -5.6005 \end{bmatrix}$

Question 5

We will estimate the MA-q system output $x_{est} = v[k] * \hat{h}[k]$ and calculate the normalized root mean square error (NMRSE)

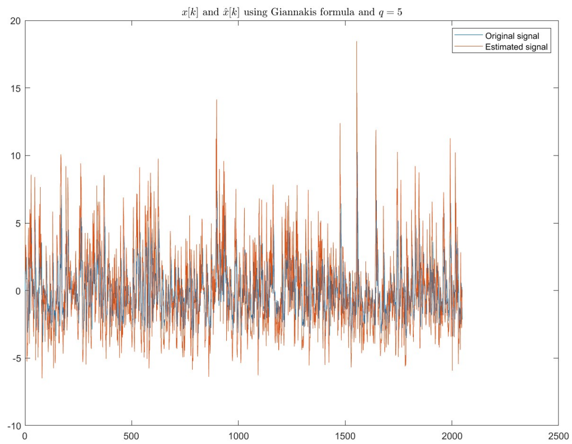

Figure 2: Plot of original and estimated signal using Giannakis' formula (q=5)

The NRMSE using the correct order of the system is equal to 0.2164. This value indicates a fairly good estimation accuracy, even though the coefficients in $\hat{h}[k]$ are different from the coefficients of the process x[k] we initially generated. This also appears clearly on the plot of figure 2, where the estimated signal seems to mostly follow the original.

Question 6

In the same manner as question 5, the MA-q system output is estimated, this time in the cases of sub-estimation and sup-estimation of the order q

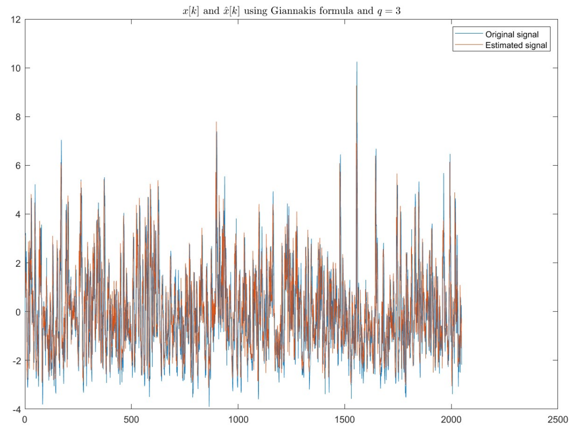

Figure 3: Plot of original and estimated signal using Giannakis' formula (q=3)

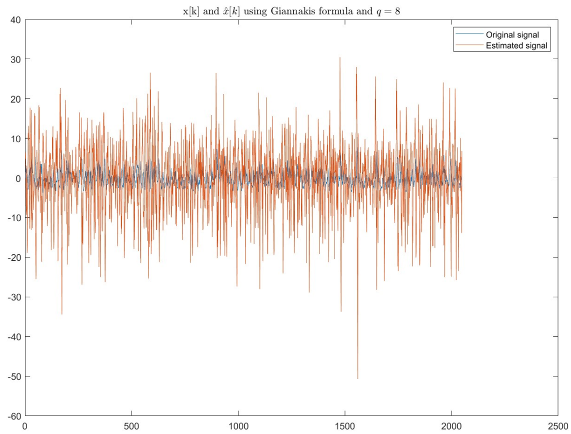

Figure 4: Plot of original and estimated signal using Giannakis' formula (q=8)

The NRMSE for $q = 3$ is equal to 0.1328, while for $q = 8$ it is 0.6517. We immediately notice that the NRMSE considering a sub-estimation of the system's order q is lower than the NRMSE we calculated previously with the accurate order q. This indicates that Giannakis' Formula actually performs better on MA systems of sub-estimated order, which is apparent on Figure 3 as well. It also appears, both on the NRMSE and the plot, that the performance is a lot worse when dealing with a sup-estimation of the order q.

Question 7

We repeat the previous steps, this time contaminating the signal with white Gaussian noise at the output. For SNR values ranging from -30dB to -5dB, we calculate and plot the NRMSE error, in order to examine the robustness of Giannakis' Formula with regards to additive noise.

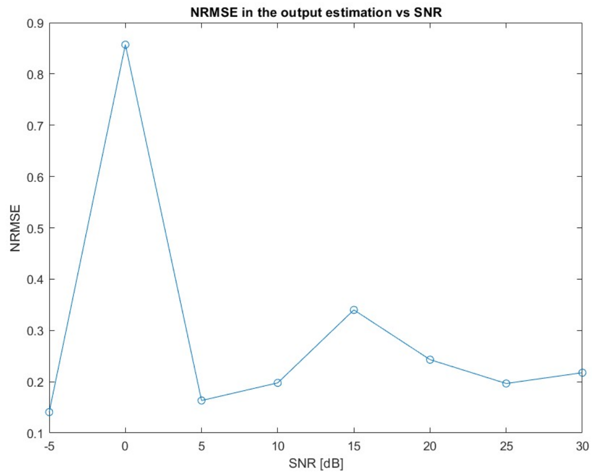

Figure 5: Plot of NRMSE over several SNR values

The NRMSE should remain generally unaffected by the presence of additive white gaussian noise, as the 3rd order cumulants are not impacted by this change, staying in low values. On the above plot, however, we notice inconsistencies to this statement, with the NRMSE even peaking at a value of 0.87 for SNR=0 (equal contribution of signal and noise). This error can be attributed to the fact that a finite set of samples is used when calculating the 3rd order cumulants.

Question 8

In order to get to a more general conclusion when it comes to the behaviour of the system's NRMSE, we will repeat the whole process 50 times and extract results using the mean values of our calculations. The results are as follows:

NRMSE for $q = 5$: $mu = 0.3980$ $std = 1.2350$
NRMSE for $q = 3$: $mu = 0.1500$ $std = 0.0173$
NRMSE for $q = 8$: $mu = 0.7395$ $std = 1.6135$

Our initial observation that Giannakis' Formula performs best when the system's order is sub-estimated is validated.

When it comes to performance with additive noise, the following plot of the mean NRMSE over the SNR of 50 iterations, shows a more robust response, as expected.

It must be noted that in all of the previous cases, and even under multiple iterations, the results show inconsistencies, with the plot differing each time we run the code. There are even some times when the NRMSE rises way above 1, such as in the plot below.

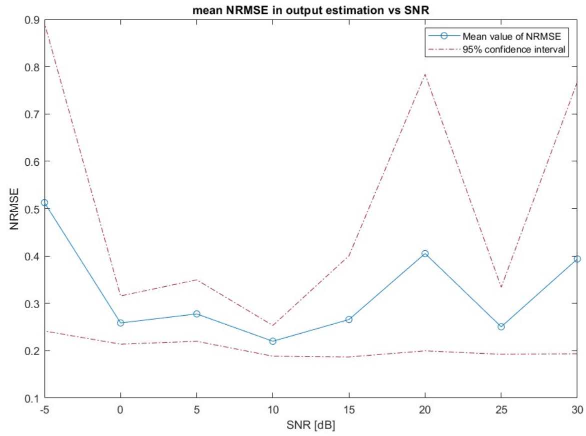

Figure 6: Plot of mean NRMSE over several SNR values

We can conclude, that Giannakis' Formula is most effective in the theoretical case of infinite available samples of the signal, not so in a practical example such as the one this exercise addresses.

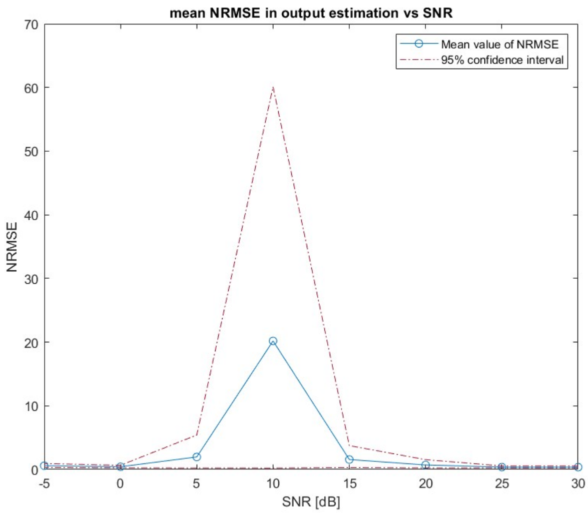

Figure 7: Mean NRMSE vs SNR - Flawed result

1 Appendix: MATLAB Codes

Includes the matlab files exercise3.m , X_Signal.m, NRMSE_calc.m and GiannakisFormula.m The functions bisp3cum, lagwind and toep which were needed, were taken from MATLAB File Exchange and are included in the project's .zip folder.

```
% Homework Exercise 3:
% Validity check of Giannakis' formula
% Dafni Nikolaidou   10546

clc;
clear;
close all;

N = 2048;     % number of samples
q = 5;        % order of MA system

% MA process coefficients
b = [1.0, 0.93, 0.85, 0.72, 0.59, -0.1];

% Signal X construction
%[x,v] = X_Signal(N,b,q);  % uncomment to create a different X

load('X_signal.mat')
```

Q 1: Justify the non-Gaussian character of v (skewness calculation)

```
mu_v = mean(v);
std_v = std(v);
skewness_v = sum((v-mu_v).^3)/((N-1)*std_v^3);
fprintf('The skewness of v[k] is: %.4f\n', skewness_v);
% Not equal to zero thus v is non-Gaussian
```

Q 2: Estimation and plots of 3rd-order cumulants of x (indirect method K=32, M=64, L=20)

```
K = 32;
M = 64;
L = 20;

[~,~,cum3_x,~] = bisp3cum(x,M,L,'n','u');

% Plots of 3rd order cumulants
subplot(1,2,2)
axis=-L:L;
surf(axis,axis,cum3_x);
title('3rd order cumulants of x[k]');
xlabel('\tau_1'); ylabel('\tau_2');

subplot(1,2,1)
contour(axis,axis,cum3_x);
title('3rd order cumulants of x[k]');
xlabel('\tau_1'); ylabel('\tau_2');
```

Q 3: Estimate the impulse response of the MA system from Giannakis's formula

```
h = GiannakisFormula(cum3_x,q,L);
fprintf("Estimated impulse response of the MA system - Giannakis' formula" + ...
    " (order q=%d):\n",q);
disp(h);
```

Q 4: Estimation of h[k] from Giannakis's formula for sub-estimation and sup-estimation of the order q

```
% Sub-estimation of the order q
q_sub = q - 2;
h_sub = GiannakisFormula(cum3_x,q_sub,L);
fprintf("Estimated impulse response - Giannakis' formula" + ...
    "(q=%d):\n",q_sub);
disp(h_sub);

% Sup-estimation of the order q
q_sup = q + 3;
h_sup = GiannakisFormula(cum3_x,q_sup,L);
fprintf("Estimated impulse response - Giannakis' formula" + ...
    "(q=%d):\n",q_sup);
disp(h_sup);
```

Q 5: Estimate MA-q system output, plot original and estimated x[k] and find the NRMSE

```
x_est = conv(v,h,'same')';

figure();
plot(1:N,x);
hold on;
plot(1:N,x_est);
title('$x[k]$ and $\hat{x}[k]$ - Giannakis formula and $q=5$','Interpreter', 'Latex');
legend('Original signal','Estimated signal');

nrmse = NRMSE_calc(x,x_est,N);
fprintf('NRMSE for q = %d: %.4f\n',q,nrmse);
```

Q 6: Repeat Q 5 for h_sub and h_sup

```
% Sub-estimation of order q
x_est_sub = conv(v,h_sub,'same')';

figure();
plot(1:N,x);
hold on;
plot(1:N,x_est_sub);
title('$x[k]$ and $\hat{x}[k]$ - Giannakis formula and $q=3$', 'Interpreter', 'Latex');
```

```
legend('Original signal','Estimated signal');

nrmse_sub = NRMSE_calc(x,x_est_sub,N);
fprintf('NRMSE for q = %d: %.4f\n',q_sub,nrmse_sub);

% Sup-estimation of order q
x_est_sup = conv(v,h_sup,'same')';

figure();
plot(1:N,x);
hold on;
plot(1:N,x_est_sup);
title('x[k] and $\hat{x}[k]$ - Giannakis formula and $q=8$', 'Interpreter', 'Latex');
legend('Original signal','Estimated signal');

nrmse_sup = NRMSE_calc(x,x_est_sup,N);
fprintf('NRMSE for q = %d: %.4f\n',q_sup,nrmse_sup);
```

Q 7: Repeat Q 2,3,5 if we add AWGN at the output, producing a variation in SNR

```
snr = 30:-5:-5;
n = length(snr);
nrmse_awgn = zeros(1,n);

for i=1:n
    % output with noise
    y = awgn(x,snr(i),'measured');

    % 3rd order cumulants
    [~,~,cum3_y,~] = bisp3cum(y,M,L,'n','u');

    % impulse response - Giannakis' formula - q=5
    h_awgn = GiannakisFormula(cum3_y,q,L);

    % Output and NRMSE
    y_est = conv(v,h_awgn,'same')';
    nrmse_awgn(i) = NRMSE_calc(y,y_est,N);
end

% NRMSE - SNR plot
figure();
plot(snr,nrmse_awgn,'-o');
xlabel('SNR [dB]'); ylabel('NRMSE');
title('NRMSE in the output estimation vs SNR');
```

Q 8: Repeat the process 50 times and use mean values

```
r = 50;
NRMSE = zeros(1,r);
```

```
NRMSE_awgn = zeros(r,n);
NRMSE_sub = zeros(1,r); NRMSE_sup = zeros(1,r);

for i=1:r
    [x_i,v_i] = X_Signal(N,b,q);
    %3rd order cumulants
    [~,~,cum3_xi,~] = bisp3cum(x_i,M,L,'n','u');

    % Estimate impulse response from Giannakis' formula
    h_i = GiannakisFormula(cum3_xi,q,L);
    h_i_sub = GiannakisFormula(cum3_xi,q_sub,L);
    h_i_sup = GiannakisFormula(cum3_xi,q_sup,L);

    % Estimate output and NRMSE of each case
    x_est_i = conv(v_i,h_i,'same')';
    NRMSE(i) = NRMSE_calc(x_i,x_est_i,N);
    x_est_i = conv(v_i,h_i_sub,'same')';
    NRMSE_sub(i) = NRMSE_calc(x_i,x_est_i,N);
    x_est_i = conv(v_i,h_i_sup,'same')';
    NRMSE_sup(i) = NRMSE_calc(x_i,x_est_i,N);
    for j=1:n
        % output with noise
        y = awgn(x_i,snr(j),'measured');

        % 3rd order cumulants
        [~,~,cum3_y,~] = bisp3cum(y,M,L,'n','u');

        % impulse response - Giannakis' formula - q=5
        h_awgn = GiannakisFormula(cum3_y,q,L);

        % Estimate the output and find NRMSE
        y_est = conv(v_i,h_awgn,'same')';
        NRMSE_awgn(i,j) = NRMSE_calc(y,y_est,N);
    end
end
% Mean values of NRMSE
meanNRMSE = mean(NRMSE);
stdNRMSE = std(NRMSE);
meanNRMSE_sub = mean(NRMSE_sub);
stdNRMSE_sub = std(NRMSE_sub);
meanNRMSE_sup = mean(NRMSE_sup);
stdNRMSE_sup = std(NRMSE_sup);
meanNRMSE_awgn = mean(NRMSE_awgn);

fprintf(['\nMean valuse and standard deviation of NRMSE for %d ' ...
    'iterarations:\n'],r);
fprintf('\tNRMSE for q=%d: mu=%.4f std=%.4f\n',q,meanNRMSE,stdNRMSE);
fprintf('\tNRMSE for q=%d: mu=%.4f std=%.4f\n',q_sub,meanNRMSE_sub, ...
    stdNRMSE_sub);
fprintf('\tNRMSE for q=%d: mu=%.4f std=%.4f\n',q_sup,meanNRMSE_sup, ...
```

```
    stdNRMSE_sup);

% 95% confidence interval - Bootstrap
alpha = 0.05;
B = 1000;     % bootstrap samples
boot_meanNRMSE_awgn = bootstrp(B, @mean, NRMSE_awgn);
% low and upper limits of confidence interval
low_lim = floor((B+1)*alpha/2);
up_lim = B+1-low_lim;

boot_meanNRMSE_awgn_sort = sort(boot_meanNRMSE_awgn);
bci_meanNRMSE_awgn = zeros(n,2);
for i = 1:n
    bci_meanNRMSE_awgn(i,1) = boot_meanNRMSE_awgn_sort(low_lim,i);
    bci_meanNRMSE_awgn(i,2) = boot_meanNRMSE_awgn_sort(up_lim,i);
end

% Plot mean NRMSE vs SNR
figure();
plot(snr,meanNRMSE_awgn,'-o');
hold on;
plot(snr,bci_meanNRMSE_awgn(:,1),'-.','Color','#A2142F');
plot(snr,bci_meanNRMSE_awgn(:,2),'-.','Color','#A2142F');
xlabel('SNR [dB]'); ylabel('NRMSE');
title('mean NRMSE in output estimation vs SNR');
legend('Mean value of NRMSE','95% confidence interval');

------------------------------------------------------------

function [x,v] = X_Signal(N,b,q)

% Construct the real discrete signal x[k], k = 1, ..., N, derived
% as the output of a MA-q process with coefficients b, driven by
% white non-Gaussian noise v[k], from exponential distribution
% with mean value of 1.

v = exprnd(1,[1,N]);  % input: White non-Gaussian noise
v = v - mean(v);      % remove mean, better NRMSE

% Output of MA-q process
x = zeros(1,N);
 for k=1:N
   for j=0:q
     if k>j
        x(k) = x(k) + b(j+1)*v(k-j);
     end
   end
 end
end
```

```
------------------------------------------------------------

function h = GiannakisFormula(cum3,q,L)
    for k = 0:q
    h(k+1) = cum3(q+L+1,k+L+1)/cum3(q+L+1,L+1);
    end
end

------------------------------------------------------------

function nrmse = NRMSE_calc(x,x_est,N)
    rmse = sqrt(sum((x_est-x').^2)/N);
    nrmse = rmse/(max(x)-min(x));
end
```