2 OCTAVE/MATLAB Problem 2.2 - Signal Statistics

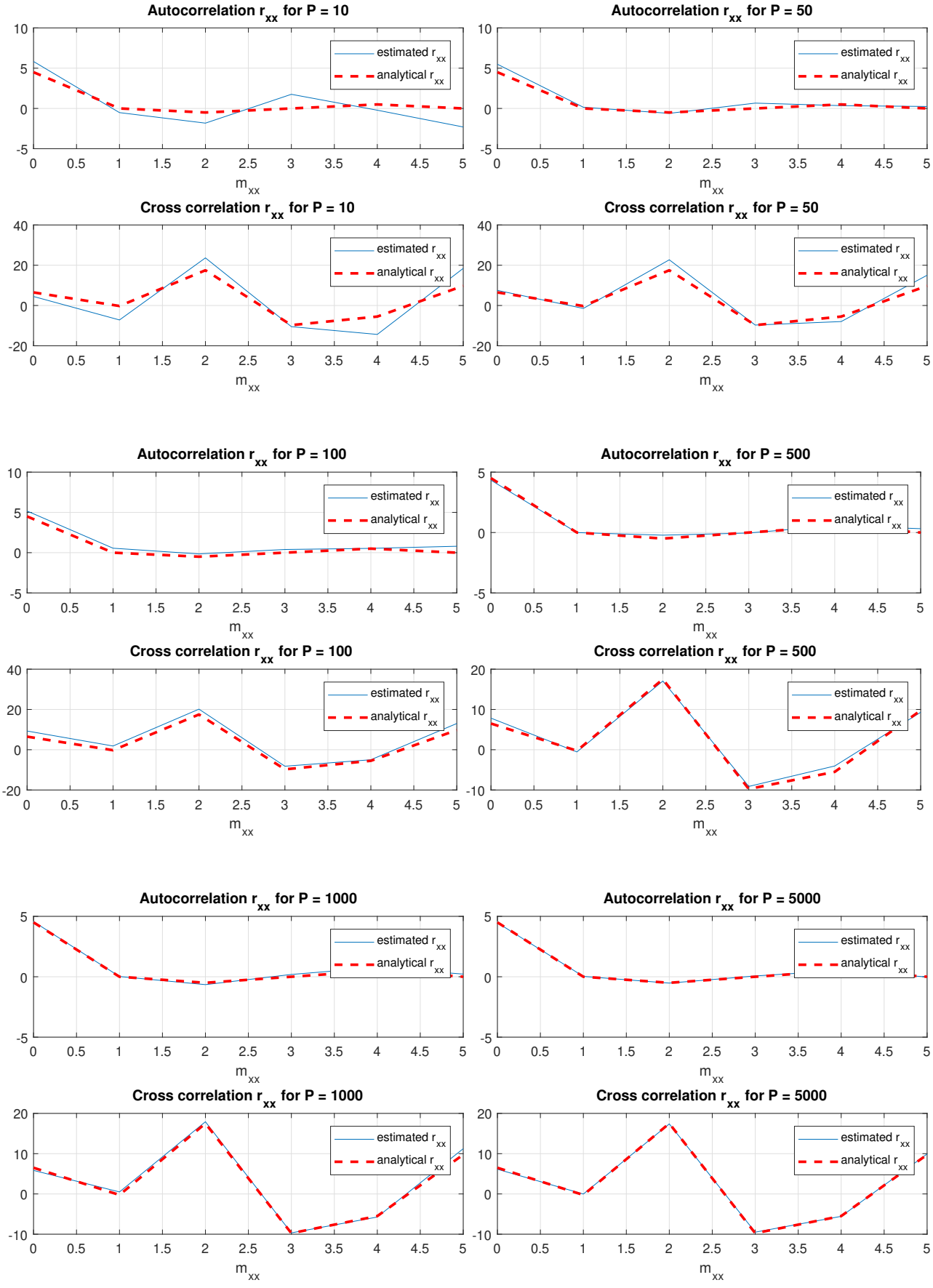
Task a)
In the first task it should be shown, that the estimated quantities are equal to the analytical within the expectation operator i.e. $\mathbf{E}\{\hat{\mu}[P]\} = \mu_x$ and $\mathbf{E}\{\hat{r}_{xy}[k,P]\} = r_{xy}[k]$ assuming $x[n]$ and $y[n]$ are jointly WSS.

$$\begin{aligned}
E\{\hat{\mu}[p]\} &= E\left\{\frac{1}{P}\sum_{n=0}^{P-1} x[n]\right\} \\
&= E\left\{\frac{1}{P}\Big(x[0] \; + \; x[1] + \; \ldots \; + x[P-1]\Big)\right\} \\
&= \frac{1}{P}\Big(E\{x[0]\} + E\{x[1]\} + \; \ldots \; + E\{x[P-1]\}\Big) \\
&= \frac{1}{P}\sum_{n=0}^{P-1} E\{x[n]\} \\
&= \frac{1}{P}\sum_{n=0}^{P-1} \mu_x \\
&= \frac{1}{P}\cdot P \cdot \mu_x \\
&= \mu_x
\end{aligned}$$

$$\begin{aligned}
E\{\hat{r}_{xy}[k,P]\} &= E\left\{\frac{1}{P}\sum_{n=0}^{P-1} x[n+k]\cdot y[n]\right\} \\
&= \frac{1}{P}\sum_{n=0}^{P-1} E\Big\{\; x[n+k]\cdot y[n] \;\Big\} \\
&= \frac{1}{P}\cdot P \cdot r_{xy}[k] \\
&= r_{xy}[k]
\end{aligned}$$

Task b)
Now a Matlab function shall be written, which estimates the auto- and cross correlation. The Matlab Code (`cross_correlation()`) can be found in the zip file and is also appended at the end of this chapter.

Task c)
The third task makes use of the just written function and calculates the correlations for different windows sizes P. The plots are given below.


Autocorrelation r_xx for P = 10
Autocorrelation r_xx for P = 50
Cross correlation r_xx for P = 10
Cross correlation r_xx for P = 50
Autocorrelation r_xx for P = 100
Autocorrelation r_xx for P = 500
Cross correlation r_xx for P = 100
Cross correlation r_xx for P = 500
Autocorrelation r_xx for P = 1000
Autocorrelation r_xx for P = 5000
Cross correlation r_xx for P = 1000
Cross correlation r_xx for P = 5000
estimated r_xx
analytical r_xx
m_xx

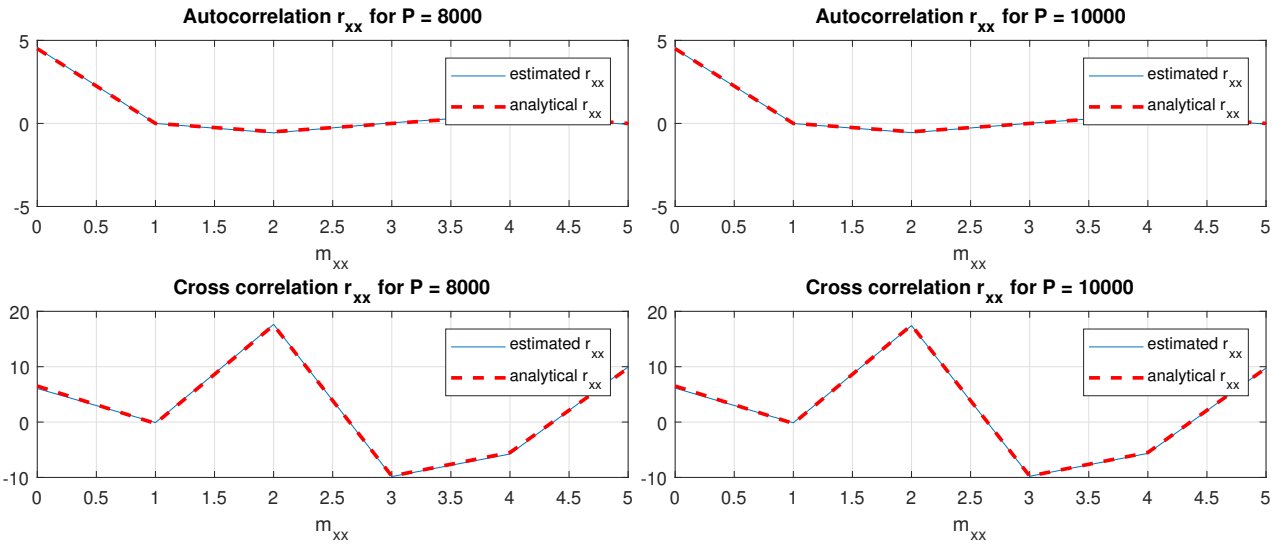


Figure 1: Auto- and cross correlation for different window sizes P

The first plot with window size $P = 10$ deviates quite a lot from the analytical solutions. The estimated correlations (which are computed by time average and not ensemble average, as the expectation operator does it) need more samples to get accurate values.

For $P = 50$ the estimated solutions look a lot better. Auto- as well as cross-correlation deviates less from the analytical solution but there is still room for improvement.

For $P = 100$ and $P = 500$ a small error can still be seen, but this error distinguishes for $P = 1000$ and above. The estimated and analytical solutions look identical.

Hence, with increasing window size P, the accuracy of the estimator increases. This was validated for several different realizations, which are not depicted in this document, but can easily be checked by running the Matlab code.

2.1 Matlab Code

Main File

```
close all
clear all
clc

mkdir Figures

load data.mat

%analytical solution
h = [2, -0.5, 4, -2, -1, 2].';

A = 1;
theta = pi/2;
sigma_u = sqrt(4);

rxx_ref = @(k) A^2/2 * cos(theta*k) + sigma_u^2 * kroneckerDelta(sym(k));
Rxx_ref = double(toeplitz(rxx_ref(0:length(h)-1))); %double convertes the sym
    data to double precision numbers

p_ref = Rxx_ref * h; %c_MSE = h = Rxx^-1 * p


%estimated solution
N = length(h);

for ii = 1:2% amount of different realizations to be plottet

 RNG = round(rand()*size(X,1)); %for each iteration choose a random realization

  for P = [10, 50, 100, 500, 1000, 5000, 8000, 10000]


      [rxx, mxx] = cross_correlation(X(RNG,:),X(RNG,:),P,N);
      [rdx, mdx] = cross_correlation(D(RNG,:),X(RNG,:),P,N);

      figure
         subplot(2,1,1)
              plot(mxx, rxx)
              grid on
              xlabel('m_{xx}')
              title( ['Autocorrelation r_{xx} for P = ' num2str(P)] )
              hold on
              plot(mxx, rxx_ref(mxx), '--r', 'LineWidth', 2);
              hold off
              legend('estimated r_{xx}', 'analytical r_{xx}')

         subplot(2,1,2)
              plot(mdx, rdx) %rdx = p
              grid on
              xlabel('m_{xx}')
              title( ['Cross correlation r_{xx} for P = ' num2str(P)] )
              hold on
              plot(mdx, p_ref, '--r', 'LineWidth', 2);
              hold off
```

```
            legend('estimated r_{xx}', 'analytical r_{xx}')


        if ii == 1   %only save for first realization
            saveas(gcf,['Figures/P=' num2str(P)], 'epsc')
        end

 end
end

 %With increasing window size P, the estimated values for rxx and rdx get
 %closer and closer to the analytical solution

 function saveas(~,~,~)
    disp('Figure not saved')
 end
```

Function cross_correlation()

```
function [rdx, mxx] = cross_correlation(x,y,P,N)


if N > P
    error('samples to average P must be greater or equal to filter coefficients N')
end

x_pad = [x(:); zeros(N-1,1)];
y = y(:); %make sure its a col vector

P_window = 1:P;

for k = 0:N-1
    rdx(k+1) = x_pad(P_window + k).' * y(P_window) / P;
end

mxx = 0:N-1;
rdx = rdx(:); %make sure its a col vector


end

% [rdx, mxx] = cross_correlation([1 2 3 4 5],[10 20 30 40 50],3, 2)
```