Carnegie Mellon DSP Electrical & Computer ENGINEERING

Digital Signal Processing (18-491/18-691)
Spring Semester, 2024

MATLAB Solutions to Problem Set 3

MATLAB Problems

Table of Contents

Solutions to 18-491/692 Problem C3.1

written over the years by the DSP TAs

```
clear;
close all;

[x, fs] = audioread("welcome16.wav");
xe = audioread("xe.wav");
xe2 = audioread("xe2.wav");
```

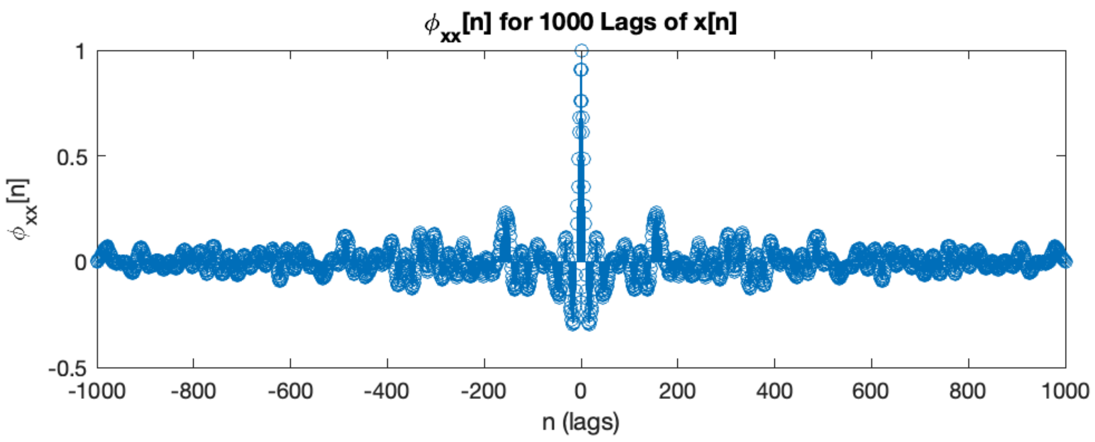
Part (a)

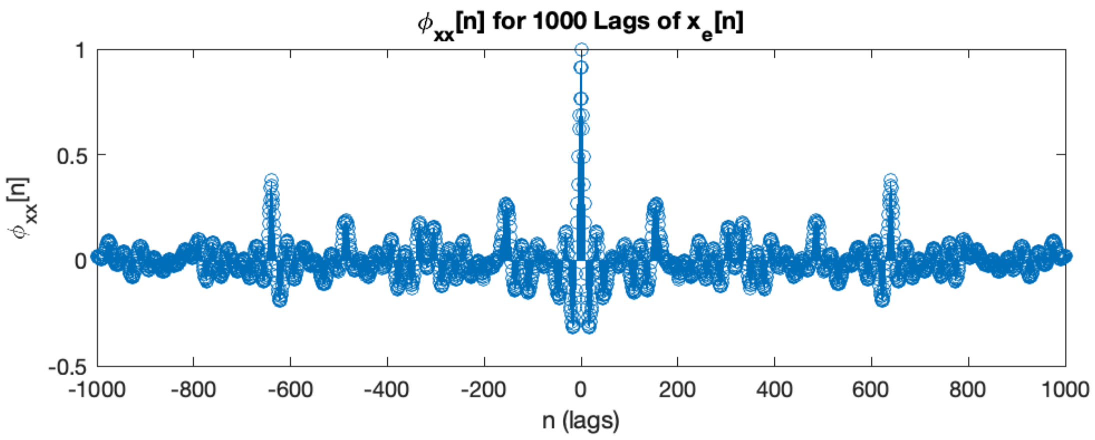
```
N = round(.04*fs);
alpha = 0.4;
h = [1; zeros(N-1,1); alpha];
xr = filter(1, h, xe); % Convolve with inverse filter to 'undo' echo

% Play audio to evaluate performance of this inverse filtering method
        soundsc(x, fs);
        pause;
        soundsc(xe, fs);
        pause;
        soundsc(xr, fs);
        pause;
```

Part (b), sections 1 and 2

```
[c, lags] = xcorr(x, 1000, 'normalized');
c1 = xcorr(xe, 1000, 'normalized');
figure;
subplot(211);
stem(lags, c);
title('\phi_{xx}[n] for 1000 Lags of x[n]');
xlabel('n (lags)');
ylabel('\phi_{xx}[n]');

subplot(212);
stem(lags, c1);
title('\phi_{xx}[n] for 1000 Lags of x_e[n]');
```

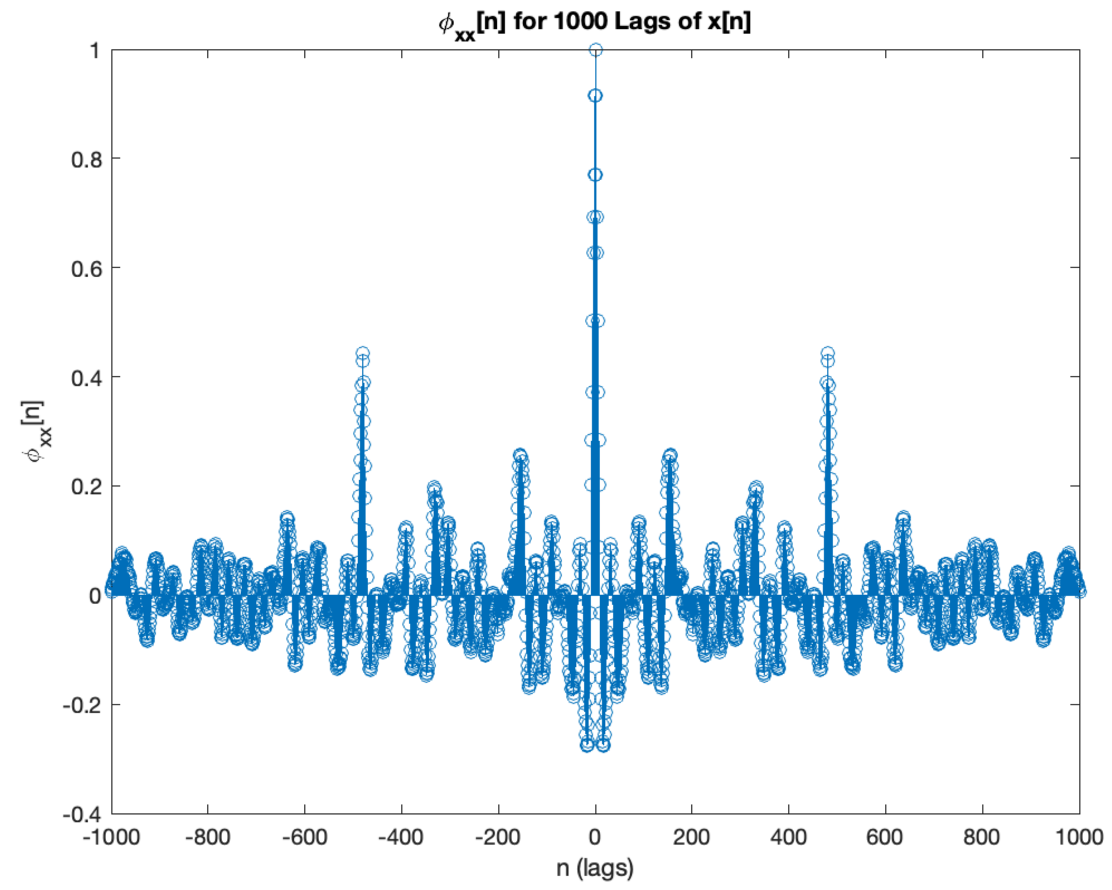
```
xlabel('n (lags)');
ylabel('\phi_{xx}[n]');
```





Part (b), sections 3 and 4

```
c2 = xcorr(xe2, 1000, 'normalized');
figure;
stem(lags, c2);
title('\phi_{xx}[n] for 1000 Lags of x[n]');
xlabel('n (lags)');
ylabel('\phi_{xx}[n]');

[n, x_corr_trunc, lags_trunc] = zero_crossing(c2);
c2_sorted = sortrows([x_corr_trunc, lags_trunc], 'descend');
alpha_est = c2_sorted(1,1);
N_est = c2_sorted(1,2);
h_est = [1; zeros(N_est-1,1); alpha_est];
xr2 = filter(1, h_est, xe2); soundsc(xr2, fs);
```



Helper functions

```
function [n, x_corr_trunc, lags_trunc] = zero_crossing(x_corr)
% This function truncates the negative half and positive main lobe main
% lobe of the autocorrelation function of x. We can truncate the negative
% half because the autocorrelation function is even, and we truncated the
% positive main lobe because we only want to consider secondary promienent
% peaks which we assume are located far from n = 0.

L = length(x_corr);
x_len = (L-1)/2+1;
half_corr = x_corr(x_len:end);

n = 1;
while half_corr(n) > 0
    n = n+1;
end

x_corr_trunc = half_corr(n:end);
lags_trunc = (n:x_len)'-1;

end
```