Contents

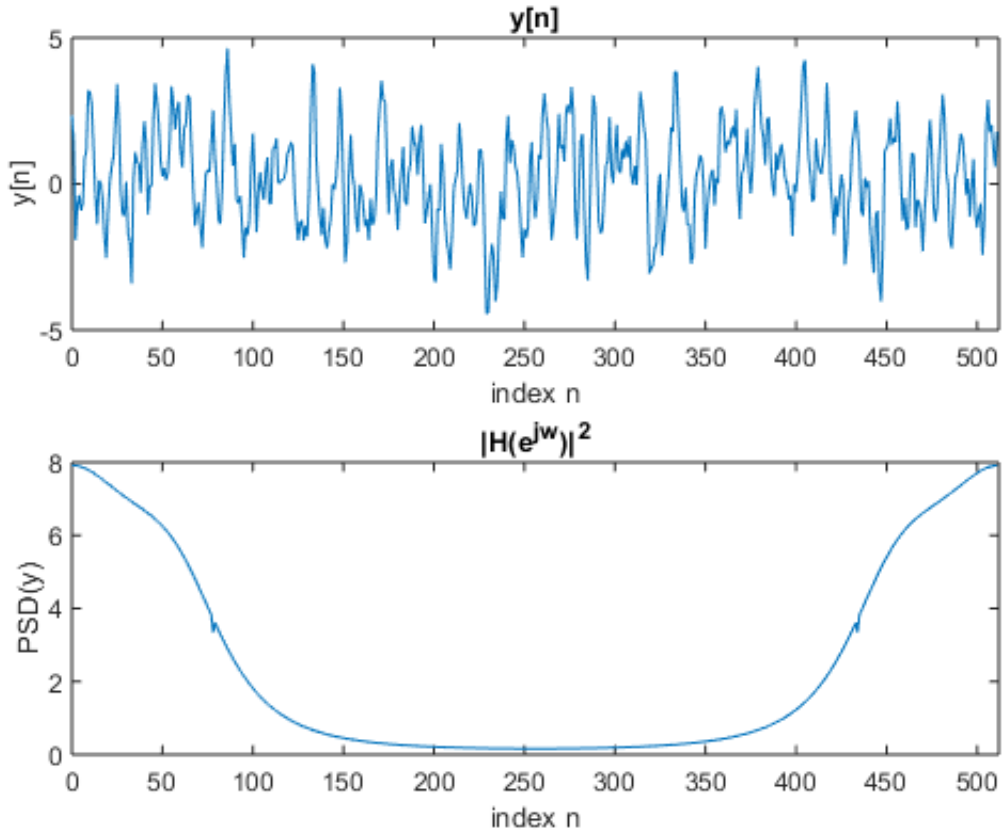

```
% Stochastics
% Andy Jeong
% 7 May 2019
%
% Project 4: Spectral Estimation
% - goal: estimate PSD of the data with coefficients a_k
% - pj2data.mat : y = data used to compute the estimate, Hejw2 = actual PSD to compare est
imates against
% - XCORR: computes an empirical estimate of autocorrelation function
% - Equation for parametric density: C-5.
% - LEVINSON: finds coefficients a_k to use to get PSD. returns a_k for p (order) = [2, 7]
% - assume variance of input x = 1, such that Syy(s) = |H(s)|^2
clc; close all; clear all;

% load data from file
data = load('pj2data.mat');
y_slice = data.y;
actualPSD = data.Hejw2;
```

Actual - y, PSD(y)

```
n = linspace(0,511,512);
k = linspace(0,511,512);
wk = 2*pi*k/512;

figure;
subplot(211); plot(n, y_slice); xlim([0,512]); title('y[n]');
xlabel('index n'); ylabel('y[n]');
subplot(212); plot(k, actualPSD); xlim([0,512]); title('|H(e^j^w)|^2');
xlabel('index n'); ylabel('PSD(y)');
```



Estimation - Comparison

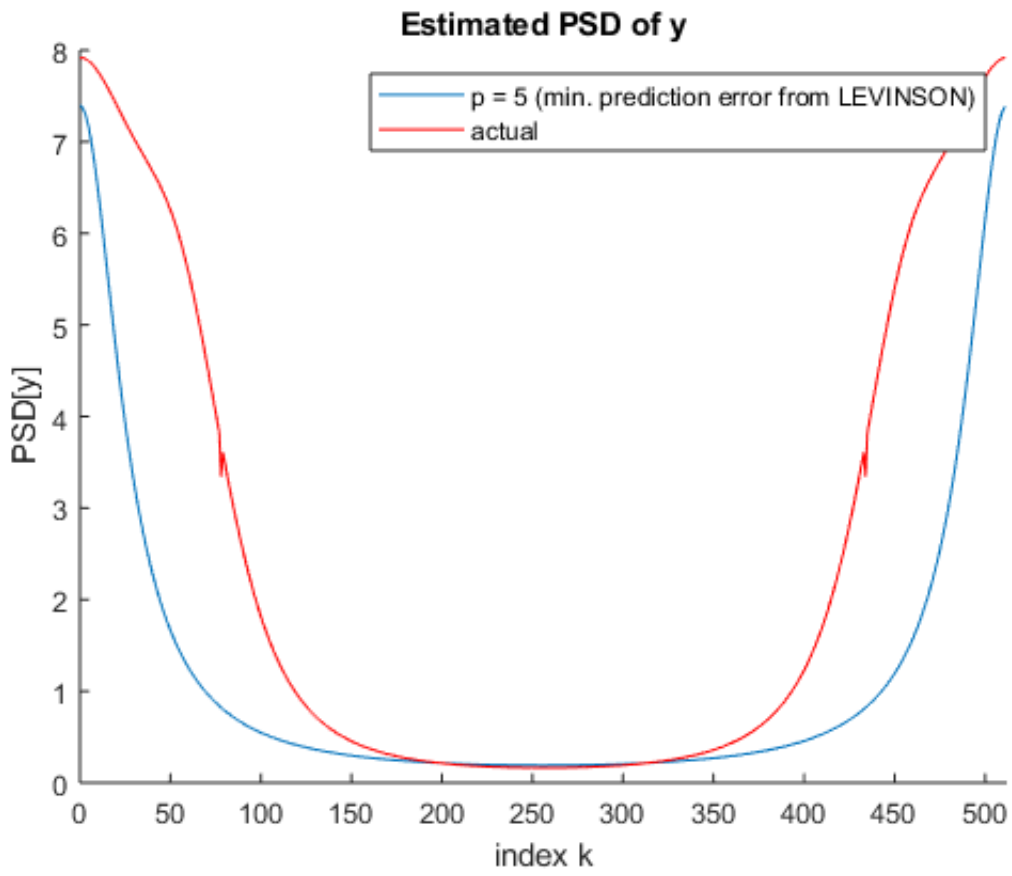
```
% coefficients and error from LEVINSON
% 'biased' divides by N, 'unbiased' divides by N-|m|
% we want 'biased' autocorrelation form
ycorr = xcorr(y_slice, 'biased');
max_p = 7;
ak_p = zeros(max_p,max_p+1); % matrix of coefficients per order p
e_p = zeros(max_p,1);        % vector of error per order p

% determine LEVINSON coeff, prediction error for PSD
for p = 2:1:max_p
    [a, e] = levinson(ycorr,p);
    ak_p(p,1:length(a)) = a;
    e_p(p) = e;
end

% Problem C.2 equation
%   - Predict minimum mean-squared value of gain factor, A
product = ones(max_p,1);
for p = 2:1:max_p
    temp = 1;
    for k = 1:1:p
        coeff = ak_p(k,k);
        temp = temp * (1-abs(coeff).^2);
    end
    product(p) = temp;
end
A = sqrt(ycorr(1).*product);        % stores gain factor per order p

% Equation C-5
%   - Frequency response H(e^jw) of a stable impulse response h
sums = zeros(max_p,512);
```

```
for p = 2:1:max_p
    sum_k = 0;
    for k = 1:1:p
        sum_k = sum_k + ak_p(p,k) .* exp(-1i*wk);
    end
    sums(p,:) = sum_k;
end
H = zeros(max_p, 512);              % initialize H
for p = 1:1:max_p
    H(p,:) = A(p) ./ (1 + sums(p,:));
end
figure; hold on;
k = linspace(0,511,512);            % index k
% plot(k, abs(H(2,:)).^2); xlim([0,512]);
% plot(k, abs(H(3,:)).^2); xlim([0,512]);
% plot(k, abs(H(4,:)).^2); xlim([0,512]);
plot(k, abs(H(5,:)).^2); xlim([0,512]);    % min prediction error: closest
% plot(k, abs(H(6,:)).^2); xlim([0,512]);
% plot(k, abs(H(7,:)).^2); xlim([0,512]);
plot(k, actualPSD,'r'); xlim([0,512]); hold off;
xlabel('index k'); ylabel('PSD[y]');
title('Estimated PSD of y');
legend('p = 5 (min. prediction error from LEVINSON)','actual')
```

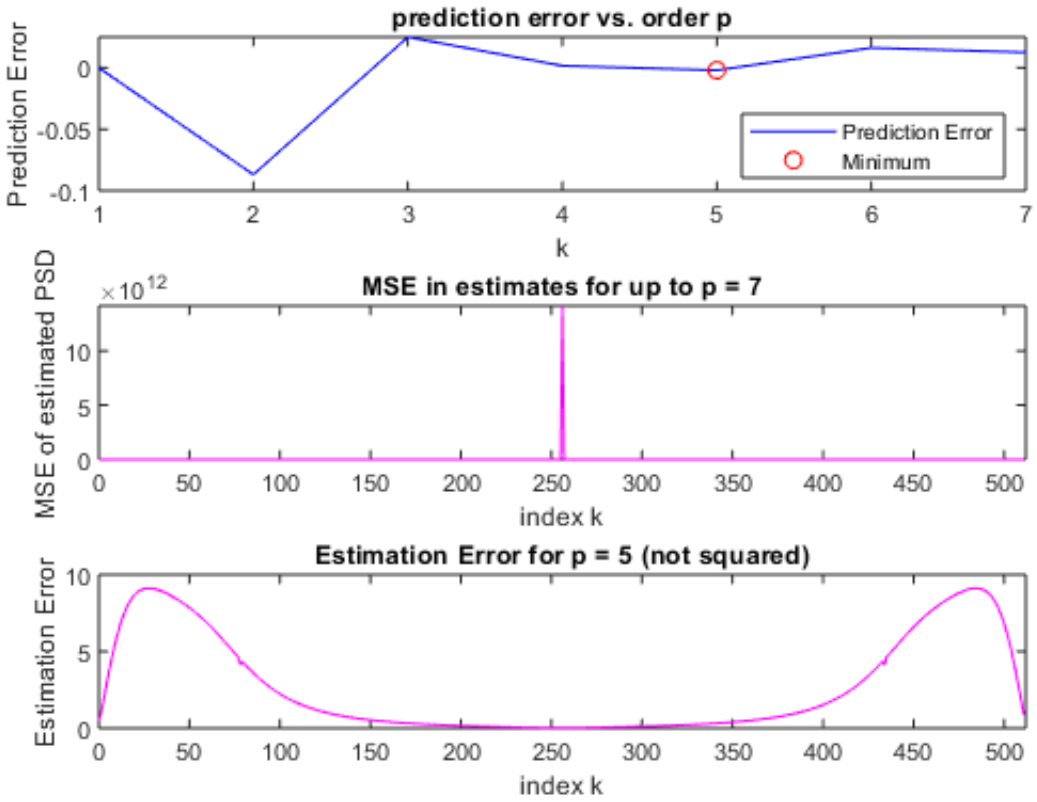


Error Plots

```
figure;
subplot(311); plot(linspace(1,7,7),e_p','b',5,transpose(e_p(5)),'ro');
xlabel('k'); ylabel('Prediction Error');
legend('Prediction Error','Minimum','Location','SouthEast');
title('prediction error vs. order p');
```

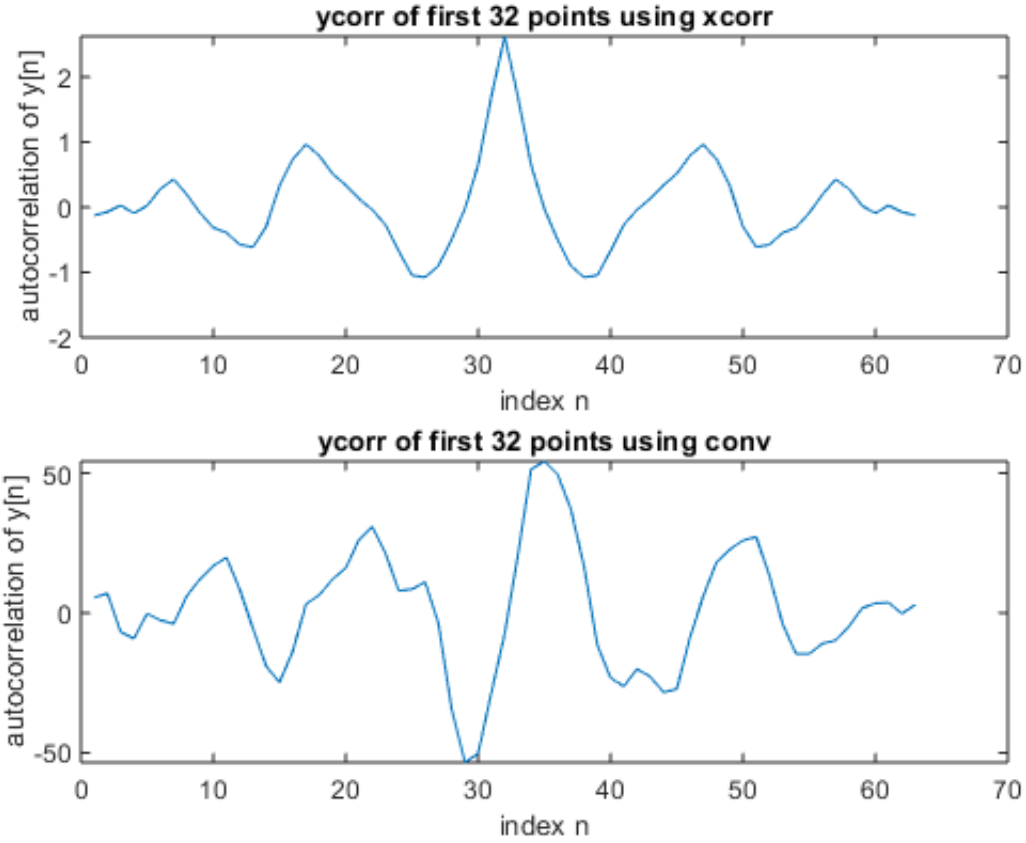
```
error = abs(H.^2-actualPSD);
mse = mean(error.^2);
subplot(312); plot(k,mse,'m');
xlabel('index k'); ylabel('MSE of estimated PSD');
title('MSE in estimates for up to p = 7'); xlim([0, 512]);

subplot(313); plot(k,error(5,:),'m');
xlabel('index k'); ylabel('Estimation Error');
title('Estimation Error for p = 5 (not squared)'); xlim([0, 512]);
```



autocorrelation of y1 = y_slice(1:32)

```
y1 = y_slice(1:32);
y1corr = xcorr(y1, 'biased');
y1corr_conv = conv(y1, y1);
n_y1 = 1:2*length(y1)-1;    % conv -> index = 2*orig_index - 1
figure;
subplot(211); plot(n_y1, y1corr); title('ycorr of first 32 points using xcorr');
xlabel('index n'); ylabel('autocorrelation of y[n]');
subplot(212); plot(n_y1, y1corr_conv); title('ycorr of first 32 points using conv');
xlabel('index n'); ylabel('autocorrelation of y[n]');
```



Published with MATLAB® R2018a