Problem 1

This problem addresses the difficulty inherent in linear prediction analysis for high-pitched speakers,

- Suppose $h[n]$ is the impulse response of an all-pole system where,

  $$H(z) = \frac{1}{1 - \sum_{k=1}^{p} \alpha_k z^{-k}}$$

  so that

  $$h[n] = \sum_{k=1}^{p} \alpha_k h[n-k] + \delta[n]$$

  Show that

  $$\sum_{k=1}^{p} \alpha_k r_h[i-k] = r_h[i]$$

- Assume $s[n]$ is a periodic waveform given by

  $$s[n] = \sum_{k=-\infty}^{\infty} h[n - kP]$$

  where P is the pitch period. Show that the autocorrelation of $s[n]$, windowed over multiple pitch periods consists of periodically repeated replicas of $r_s[\tau]$ i.e.

  $$r_s[\tau] = \sum_{k=-\infty}^{\infty} r_h[\tau - kP]$$

  but with decreasing amplitude due to the window.

- Using your results in parts (a) and (b), explain the difference between your results in part (a) and the normal equations for the autocorrelation method using the windowed speech signal $s[n]$.

- Using your results from parts (b) and (c), explain why linear prediction analysis is more accurate for low-pitched speakers than high-pitched speakers.

Solution.

$$h[n] = \sum_{k=1}^{p} \alpha_k h[n-k] + \delta[n]$$

$$h[n-i]h[n] = \sum_{k=1}^{p} \alpha_k h[n-k]h[n-i] + \delta[n]h[n-i]$$

$$\sum_{n=-\infty}^{\infty} h[n-i]h[n] = \sum_{n=-\infty}^{\infty} (\sum_{k=1}^{p} \alpha_k h[n-k]h[n-k-(i-k)] + \delta[n]h[n-i])$$

$$r_h[i] = \sum_{k=1}^{p} \alpha_k r_h[i-k]$$

(b) Given, $s[n]$ is a periodic waveform, let be a glottal pulse. We have to show that windowing $s[n]$ over multiple pitch periods would give a periodic autocorrelation function as well, with decreasing amplitude. Now, $r_h[i] = \sum_{k=-\infty}^{\infty} h[n]h[n-i]$.

$$\begin{aligned} s[n] &= \sum_{k=-\infty}^{\infty} h[n-kP] \\ \sum_{n=-\infty}^{\infty} s[n]s[n-i] &= \sum_{n=-\infty}^{\infty} (\sum_{k=-\infty}^{\infty} h[n-kP] \sum_{l=-\infty}^{\infty} h[n-i-lP]) \\ &= \sum_{k=-\infty}^{\infty} (\sum_{l=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} h[n-kP]h[n-kP-(i+(l+k)P)]) \\ &= \sum_{k=-\infty}^{\infty} r_h[i-kP] \end{aligned}$$

Linear Prediction Analysis assumes that the speech signal is based on the linear combination of past speech signal values and the contribution of the glottal pulse. Because of higher pitch, i.e. faster glottal pulse rate, deviation from the linear prediction model tends to get faster for higher-pitched voices, therefore low-pitched speakers are more accurately predicted. ■

Problem 2

Programming Assignment : (MATLAB) In this exercise, use the voiced speech signal speech1_10k (at 10000 samples/s) in the workspace ex5M1.mat. This problem illustrates the autocorrelation method of linear prediction.

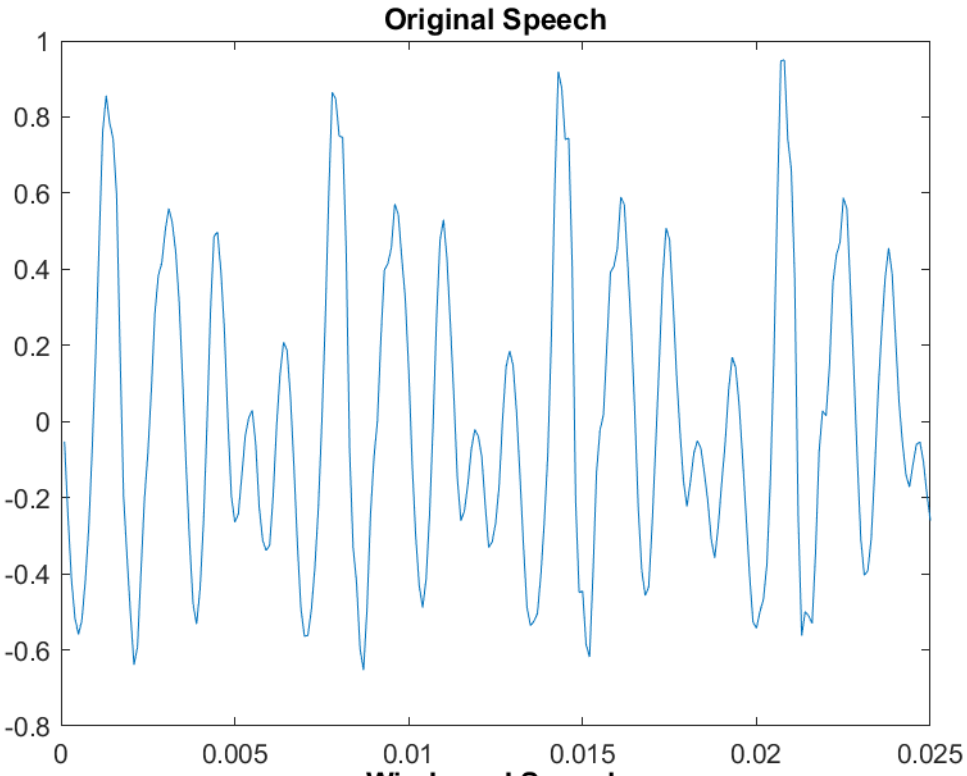
- Window speech1_10k with a 25-ms Hamming window. Compute the autocorrelation of the resulting windowed signal and plot.
- Assume that two resonances represent the signal and model the vocal tract with 4 poles. Set up the autocorrelation matrix Rn, using your result from part (a). The autocorrelation matrix is of dimension 4×4.
- Solve for the linear predictor coefficients by matrix inversion.
- Plot the log-magnitude of the resulting frequency response:

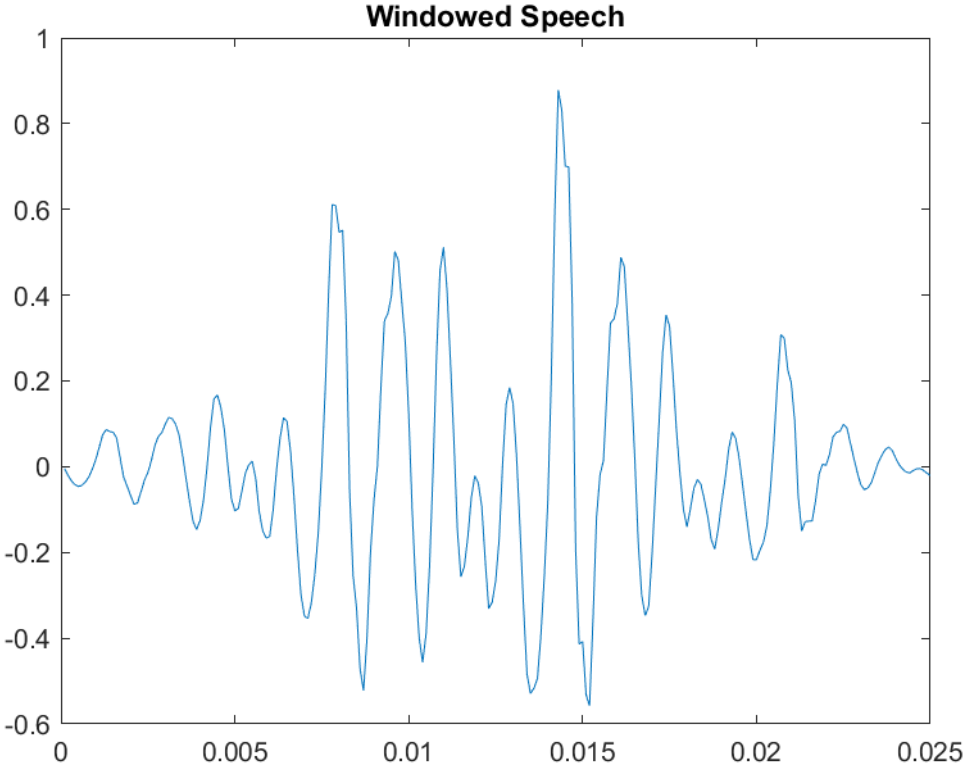
$$H(\omega) = \frac{A}{1 - \sum_{k=1}^{p} \alpha_k e^{-j\omega k}}$$

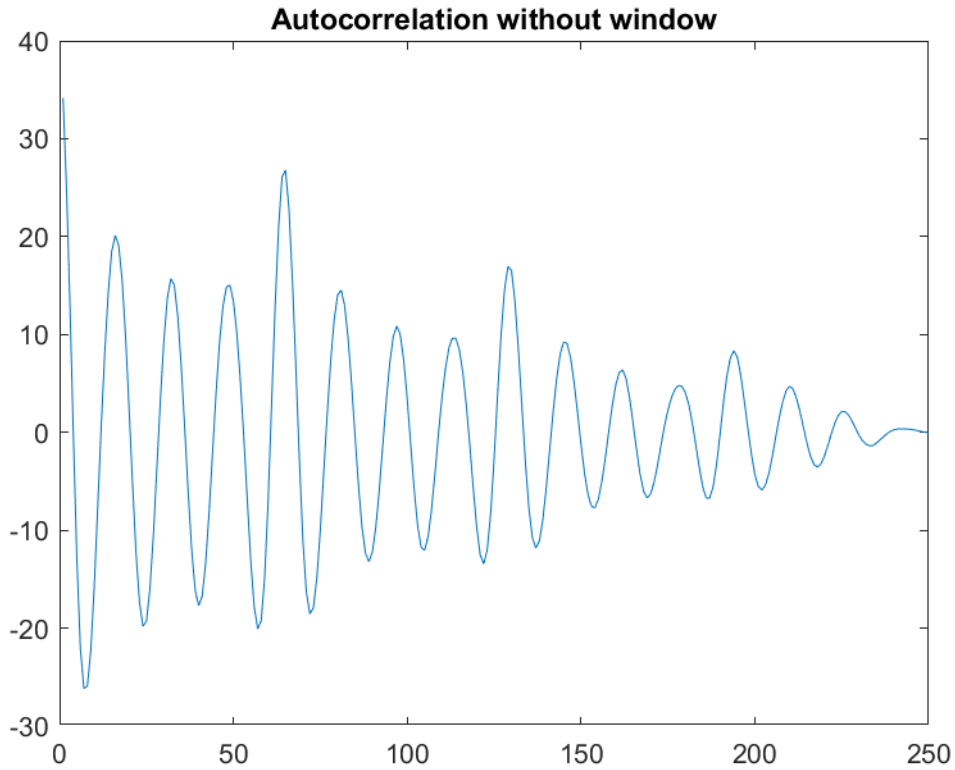
  where the gain is given by $A^2 = E_n = r_n[0] - \sum_{k=1}^{p} \alpha_k r_n[k]$. Compare your result with the log-magnitude of Fourier transform of the windowed signal. What similarities and differences do you observe?
- Using your estimates of the predictor coefficients from part (c), compute the prediction error sequence associated with *speech1_10k* and plot. From the prediction error sequence, what conclusions might one draw about the model (i.e. all-pole/impulse train driven) and estimation accuracy?

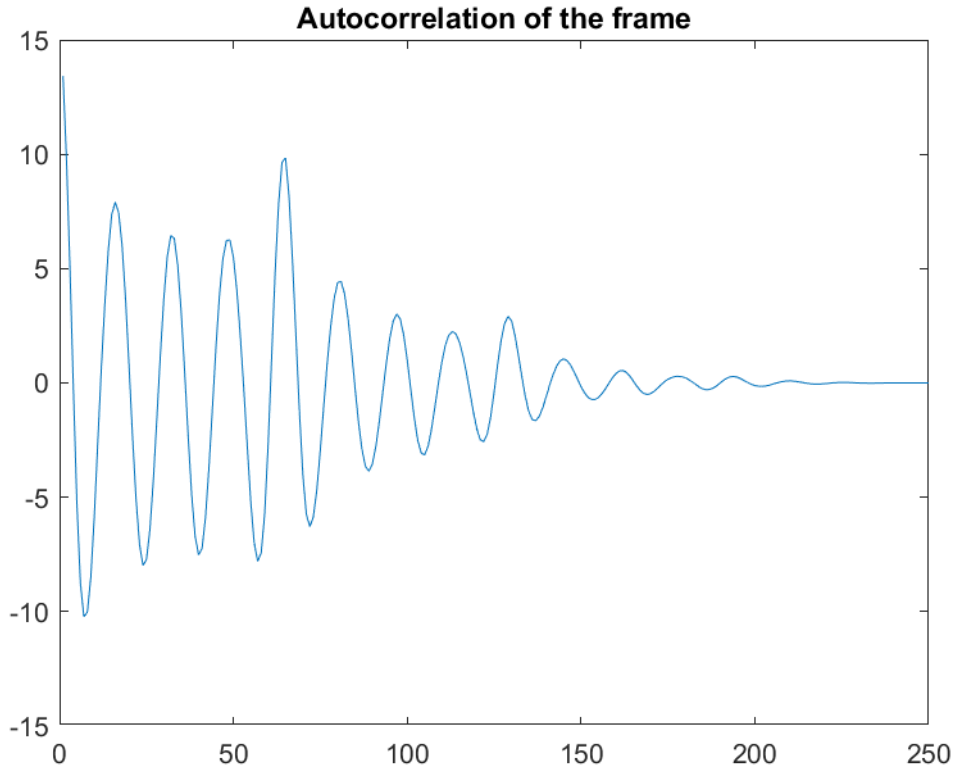
Solution.

■


Original Speech
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
0
0.005
0.01
0.015
0.02
0.025


Windowed Speech
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
0
0.005
0.01
0.015
0.02
0.025





MATLAB Code

```
[speech, Fs] = audioread('./Data_assignment_5/speech1_10k.wav');
```

```
2
3  range = (1:length(speech))/Fs;
4  plot(range, speech);
5  title('Original Speech');
6
7  window_duration = 0.025;
8  shift_duration = 0.015;
9  window = hamming(window_duration*Fs);
10 shift = shift_duration*Fs;
11
12 % Adjusting and padding speech to window
13 w_speech = zeros(1, length(window)*ceil(length(speech)/length(
       window)));
14 w_speech(1:length(speech)) = speech;
15 speech = w_speech;
16
17 num_shifts = ceil((length(speech) - length(window))/shift);
18 num_windows = num_shifts+1;
19
20 windowed_speech = zeros(num_windows, length(window));
21 for i=0:num_windows-1
22     windowed_speech(i+1, :) = speech(i*shift+1:(i*shift)+length(
          window)) .* window';
23 end
24
25 range_win = (1:length(window))/Fs;
26 % hold on;
27 for i=1:num_windows
28 %     figure;
29     plot(range_win+(i-1)*window_duration, windowed_speech(i, :));
30 %     title('Window ', i);
31 end
32 title('Windowed Speech')
33 % hold off;
34
35 plot((1:length(window))/Fs, windowed_speech(1, :));
36
37 window_autocorr = Autocorrelation(windowed_speech(1, :));
38 y = windowed_speech(1, :);
39 len = length(y);
40 y_new = zeros(len*5);
41 non_zero_range = 2*len:3*len-1;
42 y_new(non_zero_range) = y;
43 y = y_new;
44 autocorr_ = zeros(len, 1);
```

```
45 for k=1:length(autocorr_)
46     for i=non_zero_range
47         autocorr_(k) = autocorr_(k) + y(i)*y(i-k);
48     end
49 end
50 figure;
51 plot(autocorr_);
```

Problem 3

Programming Assignment : (MATLAB) In this problem you will use your results from previous problem to perform speech synthesis of the speech waveform *speech*1_10*k* in the workspace *ex5Ml.mat*.

- Using your estimates of the predictor coefficients from previous problem, compute an estimate of the vocal tract impulse response.
- Using the prediction error sequence you computed in previous problem, estimate an average pitch period of the voiced sequence speech1 10k.
- Using your results from parts (a) and (b), synthesize an estimate of speech1 10k. How does your waveform estimate differ from the original? Consider the minimum-phase nature of the impulse response estimate.
- Using the MATLAB number generator *randm.m*, synthesize the "whispered" counterpart to your voiced synthesis of part (c). Using the MATLAB sound.m function, listen to your two estimates and compare to the original.

Solution.

■

MATLAB Code