Linear Predictive Speech Synthesizer

Faiz Ahmad Khan

EEEM030

University of Surrey

Contents

[1 Introduction i](#_Toc86768688)

[1.1 Brief i](#_Toc86768689)

[2 Implementation ii](#_Toc86768690)

[2.1 Pre-processing ii](#_Toc86768691)

[2.2 Model Estimation iii](#_Toc86768692)

[2.3 Synthesis v](#_Toc86768693)

[3 Analysis vii](#_Toc86768694)

[3.1 LPC Order Variation vii](#_Toc86768695)

[3.2 Source Segment Length Variation xi](#_Toc86768696)

[3.3 Auditory Analysis xii](#_Toc86768697)

[4 Conclusion xii](#_Toc86768698)

[References xiii](#_Toc86768699)

[6 Code xiv](#_Toc86768700)

Table of Figures

[Figure 1: Input Vowel Segment ii](https://d.docs.live.net/6b5a9591954bfa8f/Documents/Linear%20Predictive%20Speech%20Synthesizer.docx#_Toc86768763)

[Figure 2: PSD & Pole Zero of Segment iii](https://d.docs.live.net/6b5a9591954bfa8f/Documents/Linear%20Predictive%20Speech%20Synthesizer.docx#_Toc86768764)

[Figure 3: Cepstrum of Segment iv](https://d.docs.live.net/6b5a9591954bfa8f/Documents/Linear%20Predictive%20Speech%20Synthesizer.docx#_Toc86768765)

[Figure 4: Synthesis of hod\_f v](https://d.docs.live.net/6b5a9591954bfa8f/Documents/Linear%20Predictive%20Speech%20Synthesizer.docx#_Toc86768766)

[Figure 5: Synthesis of heed\_m vi](https://d.docs.live.net/6b5a9591954bfa8f/Documents/Linear%20Predictive%20Speech%20Synthesizer.docx#_Toc86768767)

[Figure 6:Effect of LPC order on PSD of hod\_f viii](https://d.docs.live.net/6b5a9591954bfa8f/Documents/Linear%20Predictive%20Speech%20Synthesizer.docx#_Toc86768768)

[Figure 7:Effect of LPC order on PSD of heed\_m viii](https://d.docs.live.net/6b5a9591954bfa8f/Documents/Linear%20Predictive%20Speech%20Synthesizer.docx#_Toc86768769)

[Figure 8:Synthesis of hod\_f - LPC order = 5 ix](https://d.docs.live.net/6b5a9591954bfa8f/Documents/Linear%20Predictive%20Speech%20Synthesizer.docx#_Toc86768770)

[Figure 9:Synthesis of hod\_f - LPC order = 90 ix](https://d.docs.live.net/6b5a9591954bfa8f/Documents/Linear%20Predictive%20Speech%20Synthesizer.docx#_Toc86768771)

[Figure 10:Synthesis of heed\_m - LPC order = 5 x](https://d.docs.live.net/6b5a9591954bfa8f/Documents/Linear%20Predictive%20Speech%20Synthesizer.docx#_Toc86768772)

[Figure 11:Synthesis of heed\_m - LPC order = 90 x](https://d.docs.live.net/6b5a9591954bfa8f/Documents/Linear%20Predictive%20Speech%20Synthesizer.docx#_Toc86768773)

[Figure 12:Effect of Segment Length on PSD of heed\_m xi](https://d.docs.live.net/6b5a9591954bfa8f/Documents/Linear%20Predictive%20Speech%20Synthesizer.docx#_Toc86768774)

[Figure 13:Synthesis of heed\_m - Segment Length = 25 xi](https://d.docs.live.net/6b5a9591954bfa8f/Documents/Linear%20Predictive%20Speech%20Synthesizer.docx#_Toc86768775)

[Figure 14:Synthesis of heed\_m - Segment Length = 50 xi](https://d.docs.live.net/6b5a9591954bfa8f/Documents/Linear%20Predictive%20Speech%20Synthesizer.docx#_Toc86768776)

[Figure 15:Synthesis of heed\_m - Segment Length = 200 xii](https://d.docs.live.net/6b5a9591954bfa8f/Documents/Linear%20Predictive%20Speech%20Synthesizer.docx#_Toc86768777)

# 1 Introduction

Throughout the world complex Artificial Intelligence (AI) algorithms are being trained to model the choices and physical features of human beings. Unique physical identifiers such as facial scans and speech can now be accurately modelled by these algorithms and are already used in various commercial software. Speech analysis and recognition is an important research area in AI. This report shall deal with the analysis and recognition of key parameters that are required before the more complicated task of speech recognition.

## Brief

The objective for this report is to demonstrate the application of digital signal processing and by extension audio processing techniques to analyse and estimate a model for a given speech segment and then synthesize a speech segment from that model.

This report will be limited to the estimation and synthesis of vowels. Linear predictive coding (LPC) will be used to estimate the formant structure of vowel in each speech segment. LPC uses a linear combination of past time-domain samples to predict the current time-domain sample. (Deng & O'Shaughnessy, 2018) The frequency response will be measured and compared for various orders or LPC coefficients and segment lengths of the sound segment. Additionally, the fundamental frequency and formant frequencies will also be derived for the vowel segment.

The synthesis of vowel segment will be achieved by filtering a periodic sound source with an all-pole filter which acts as a digital vocal tract filter. This process is described as the source-filter model. (Mannell, 2020) The synthesised vowel segment will then be analysed and assessed compared to the original vowel segment. This process will be repeated for varying system parameters.

# 2 Implementation

The source-filter model was implemented using MATLAB and its various libraries and in-built functions. Additional helper functions were created to be called upon when required.

## 2.1 Pre-processing

Initially, a speech sample was loaded from the system and the required constant values for all the functions were set. Next, a vowel segment was extracted from the sample. For the purpose of this report two samples, one male and the other female were investigated, namely *heed\_m.wav* and *hod\_f.wav* respectively. The extracted vowel segments can be observed in figure 1.

For general illustration and comparison between the samples, the time segment was set to 100ms and offset at 20ms.

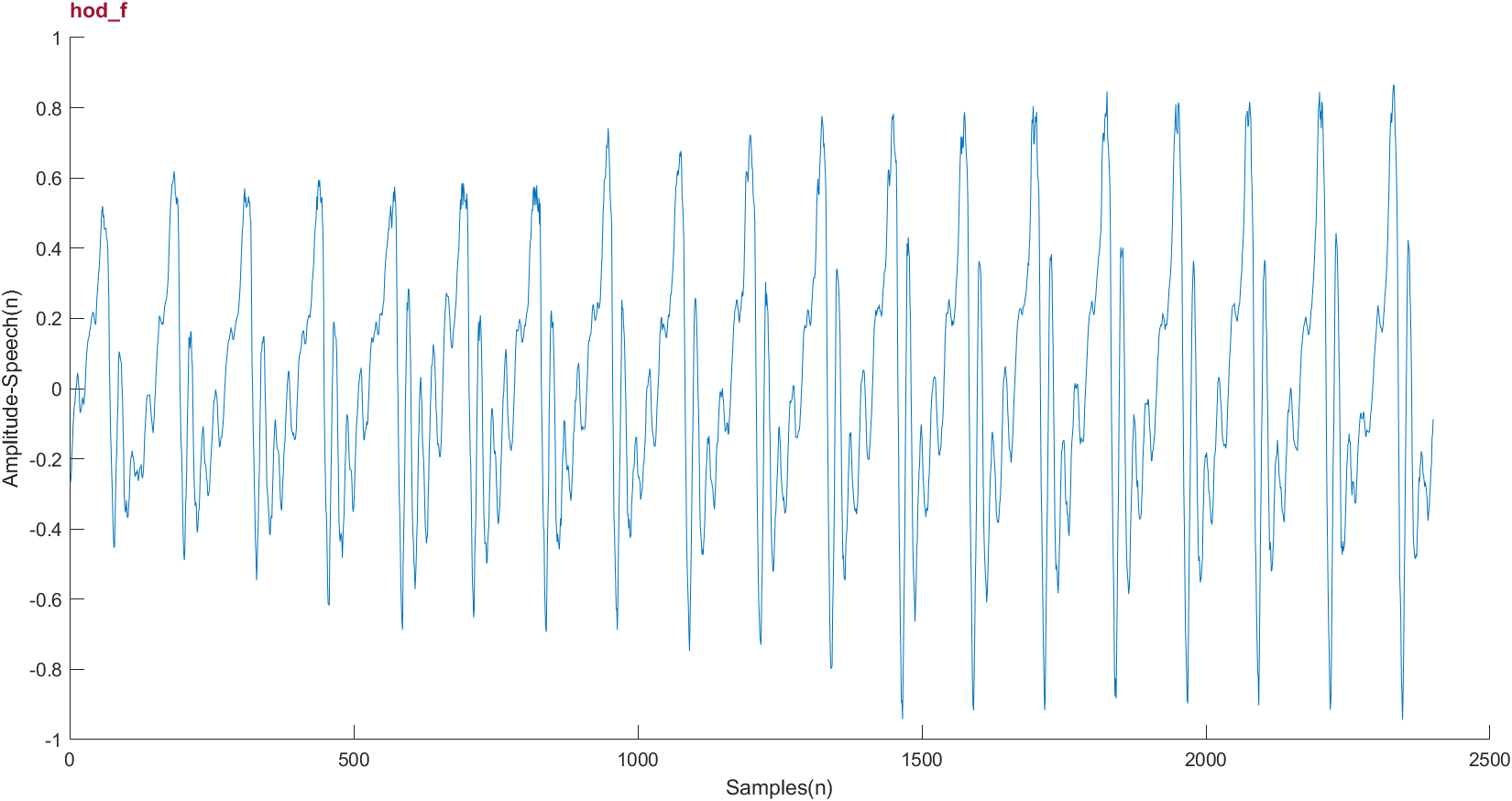
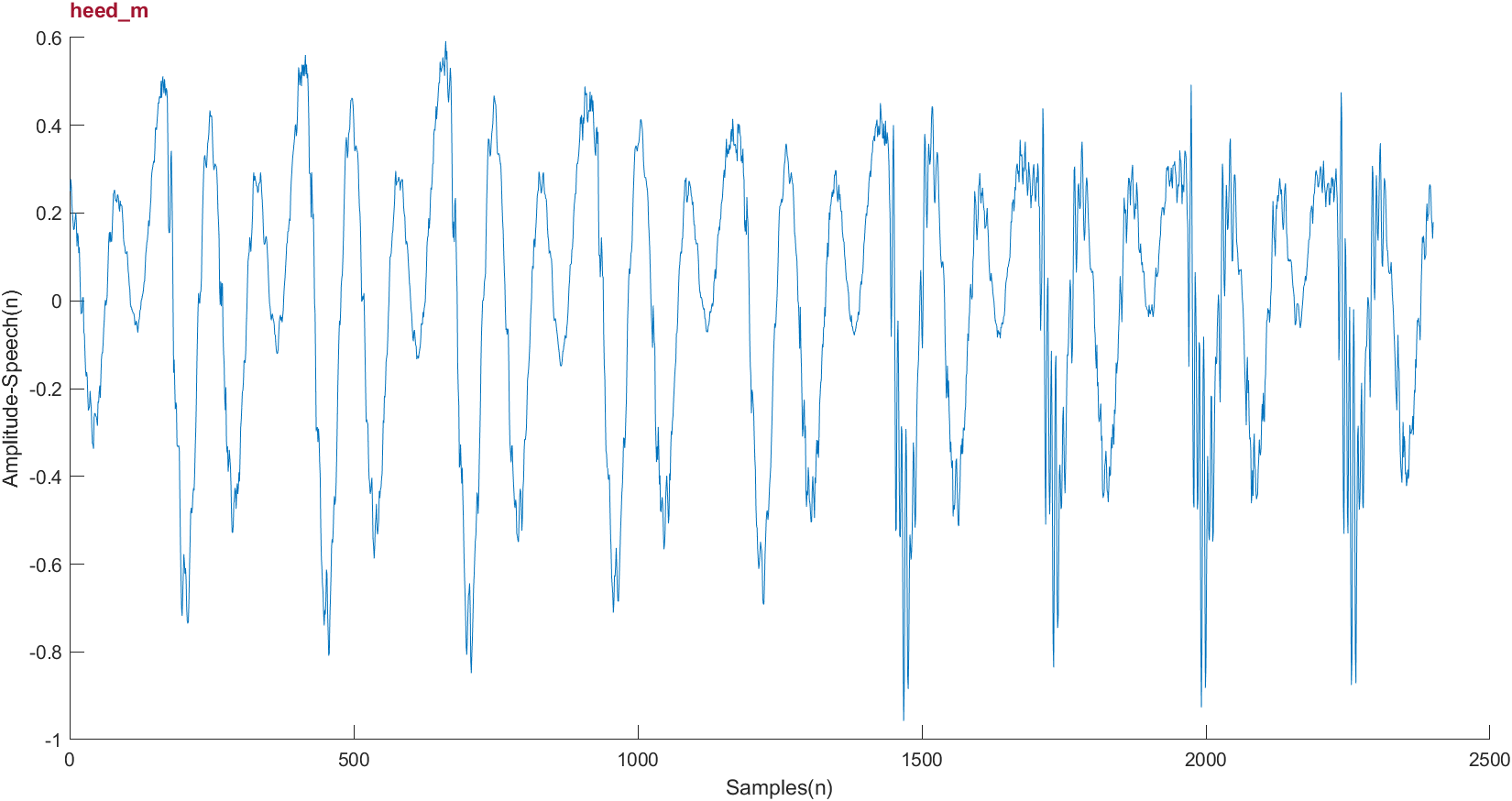


Figure 1: Input Vowel Segment

## 2.2 Model Estimation

The frequency domain representation of time domain vowel segments was computed by utilising a Fourier transform. Fourier Transform (FT) is the mathematical operation that converts time domain data into frequency domain data in the spectrum. (Yung-Li, 2005) This was then used to plot the Power Spectral Density (PSD) of the vowel segments. PSD represents the energy of the time signal at different frequencies. (Čuperlović-Culf, 2013)

Next the frequency response using the function *freqz()* was calculated for various orders of LPC coefficients. The coefficients themselves were obtained by using the function *lpc().* The frequency response of the LPC filter corresponds to the spectral envelope of the signal and the formant frequencies of the vowel manifest as the peaks of the spectrum. The peaks were calculated by passing the smooth LPC spectral envelope through a local maximum identifying function such as *islocalmax()*. Additionally, the poles and zeros of the transfer function were plotted in a pole-zero plot using the *zplane()* function. Poles and zeros give useful insights into a filter's response and can be used as the basis for digital filter design and filter stability. (Smith, 2007) The frequency response and pole-zero plot for filter order of 30 can be observed in figure 3.

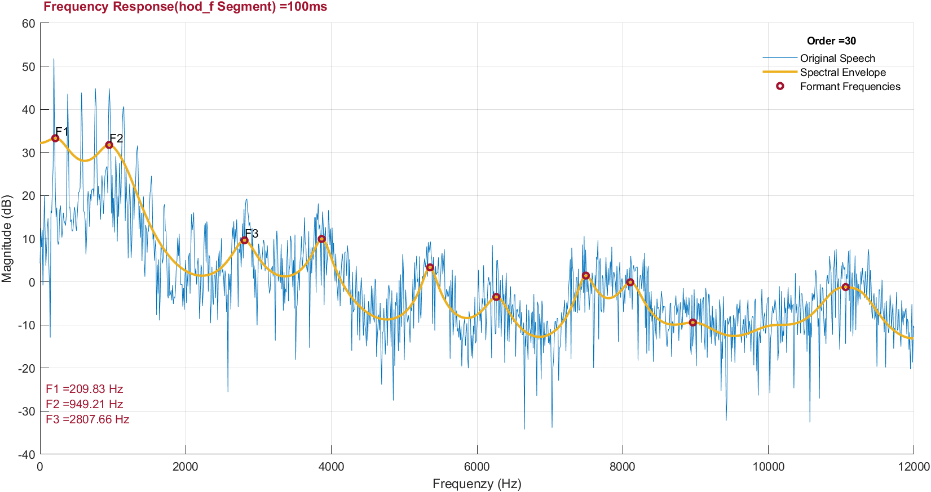
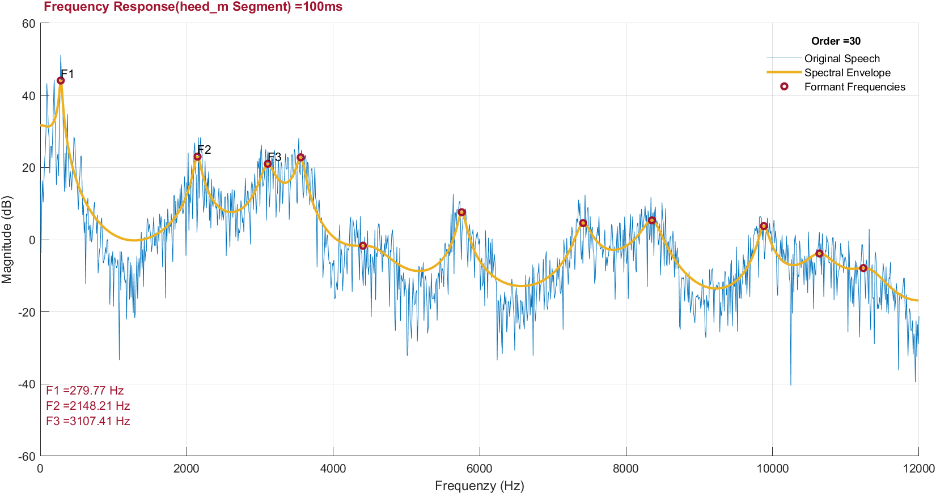
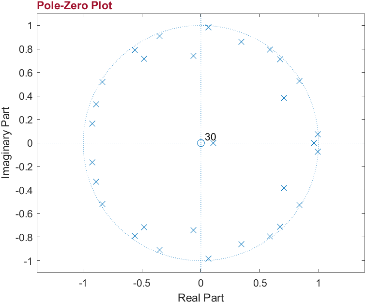


Figure 2: PSD & Pole Zero of Segment



The cepstrum of the vowel segment was plotted next to estimate the fundamental frequency. (Oppenheim & Schafer, 2004) To isolate the best candidate for fundamental frequency a sequence of steps is executed. Firstly, the mirror generated in the cepstrum plot from the use of the *rceps()* function was removed. Next the lower quefrency values defined by the quefrency threshold were removed as they represent values of frequencies that are much higher than the required fundamental frequency. The remaining values for cepstrum were passed through *islocalmax()* to find local maximum values. These values were then filtered by a cepstrum threshold value to find the best approximate fundamental frequencies of which the highest was used as pitch period. The fundamental frequency *(f0)* was then calculated by taking the inverse of the result of pitch period (*pp)* divided by the sampling frequency (*fs)*. The values of quefrency and cepstrum thresholds were set by experimenting on different values of the thresholds. These experiments were conducted on different speech segments and segment lengths to find values that would work well in all cases. The value of cepstrum threshold was set as 0.05 and quefrency threshold as 50. The cepstrum for vowel segments can be observed in figure 3.

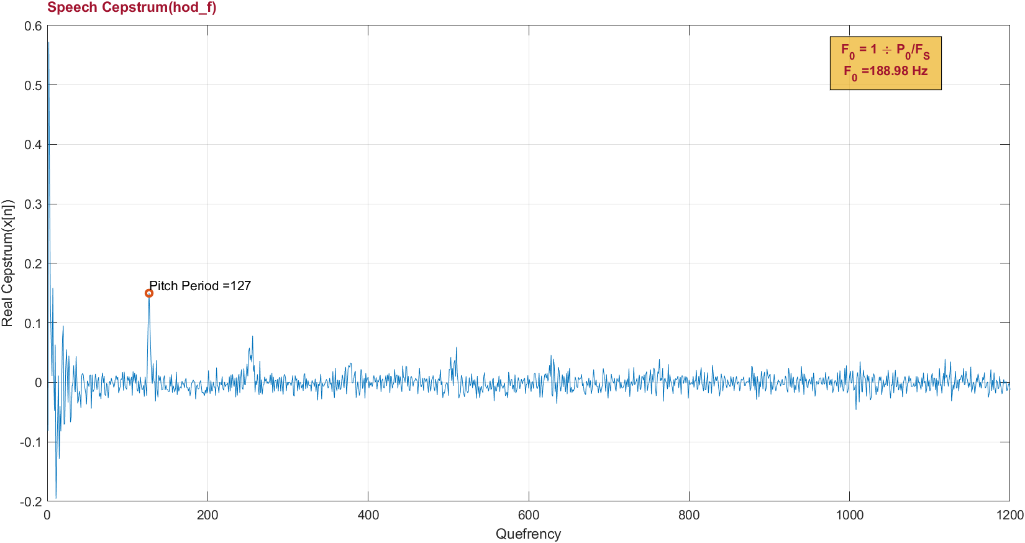
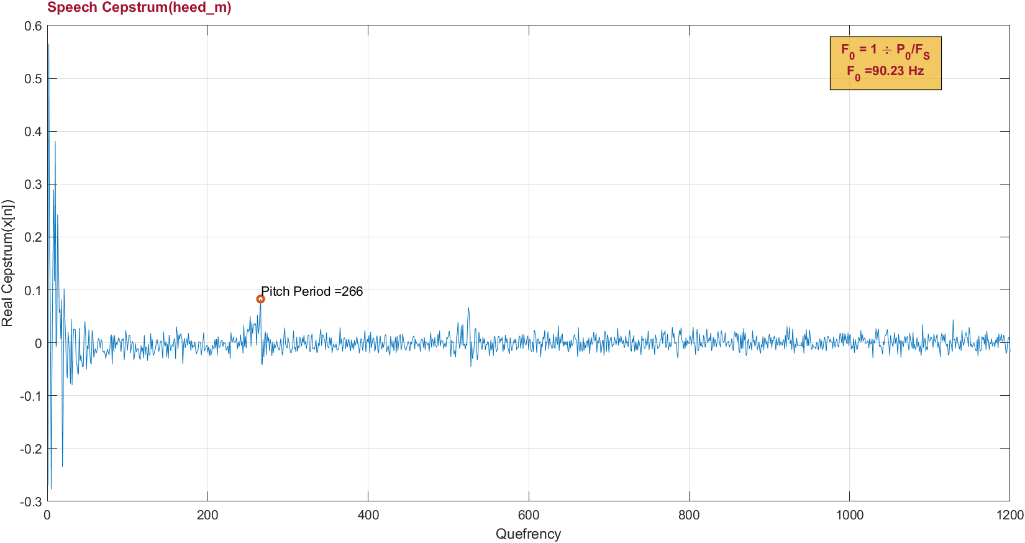


Figure 3: Cepstrum of Segment

## 2.3 Synthesis

In order to produce synthesised vowel segment, an array of the required number of samples consisting of all zeroes was generated using *zeros()* function. This array was then transformed into a periodic impulse train by setting values to one for intervals defined by the ratio of sampling frequency *(fs)* to fundamental frequency *(f0)*. The periodic impulse train was convolved with the transfer function of LPC filter to produce synthesised speech segment. This synthesised segment contains only the vowel and can be analysed by plotting the waveform and the spectrogram by using the *spectrogram()* function. The waveform and spectrogram for original and synthesised vowel were plotted with time duration constant between original and synthesised segment. This can be observed in figure 4 and figure 5 respectively.

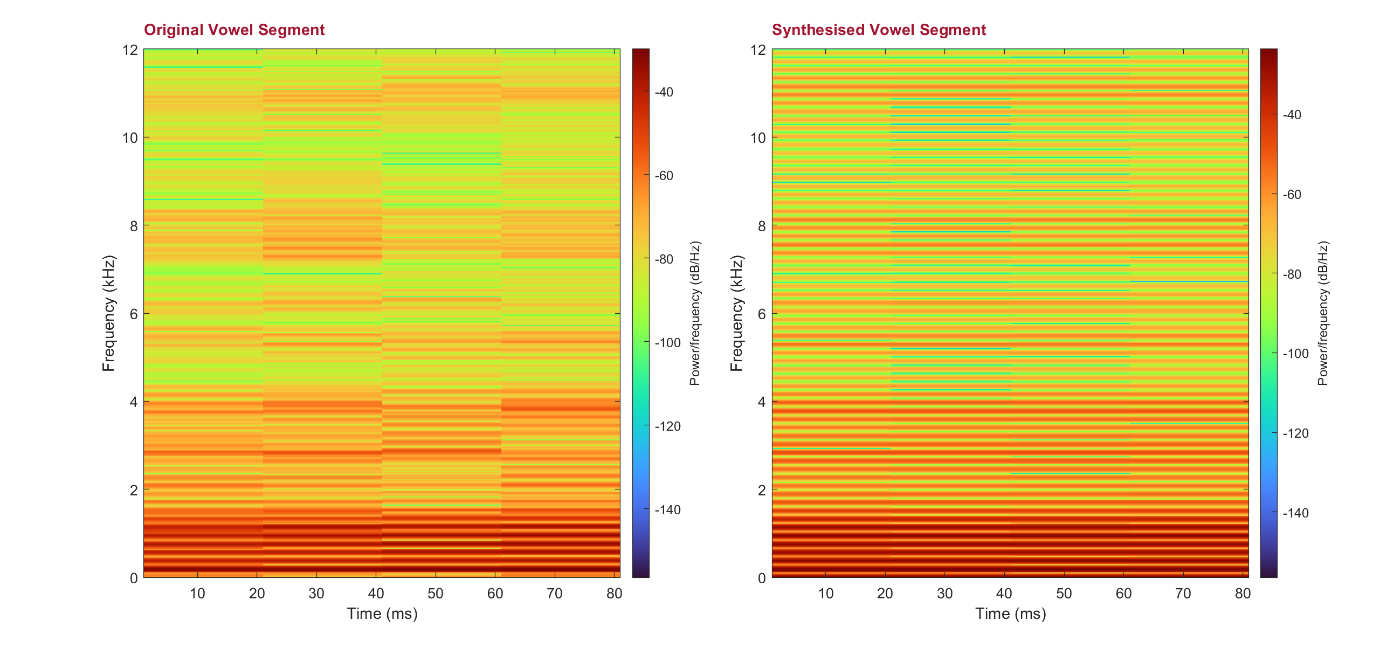
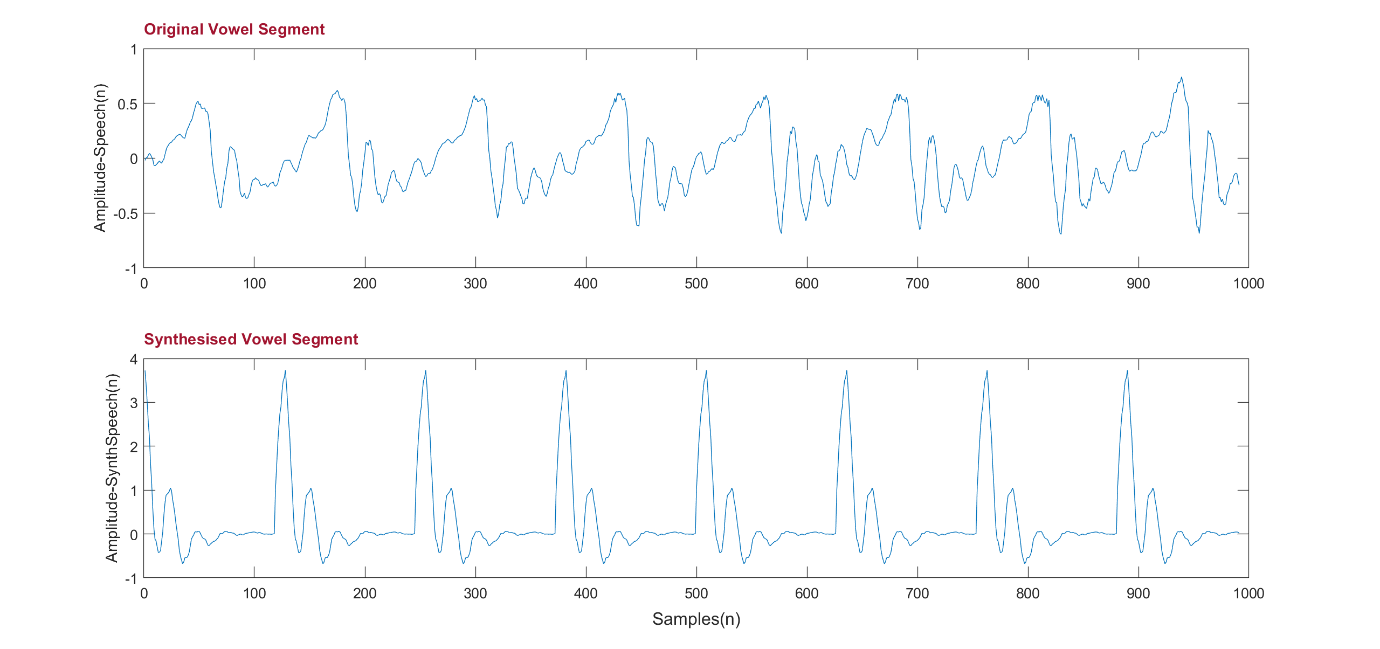


Figure 4: Synthesis of hod\_f

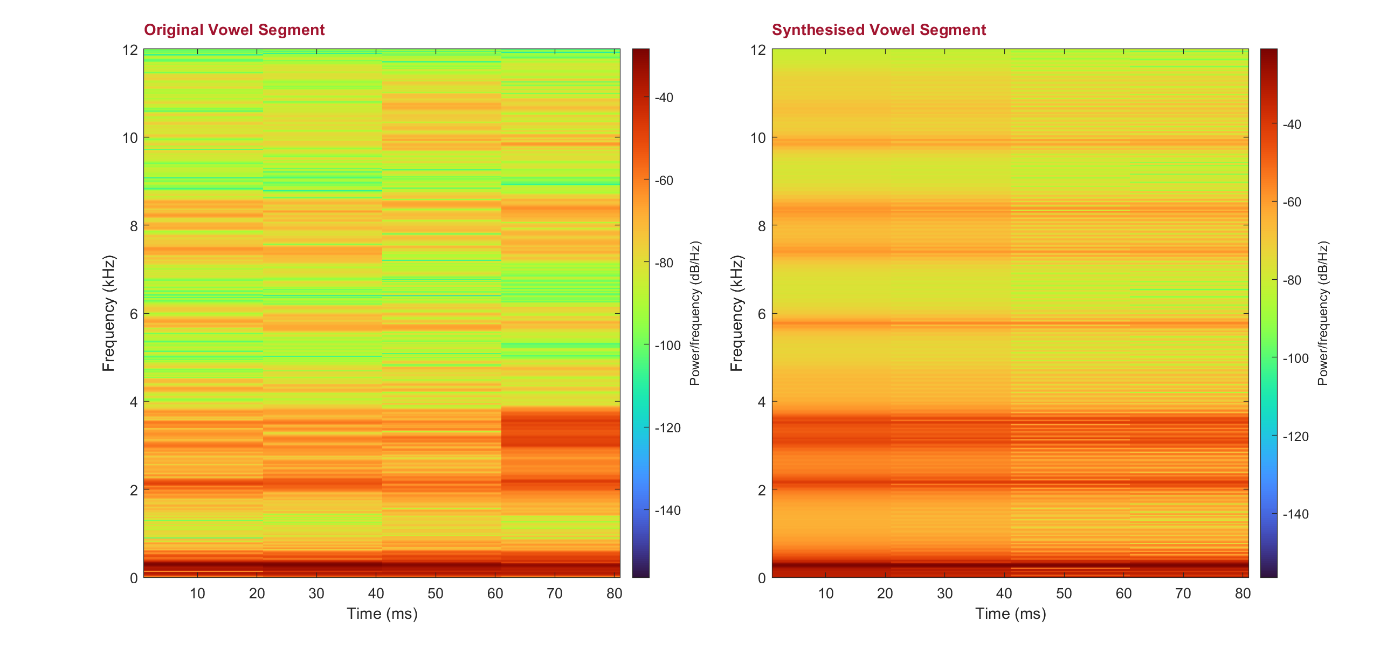
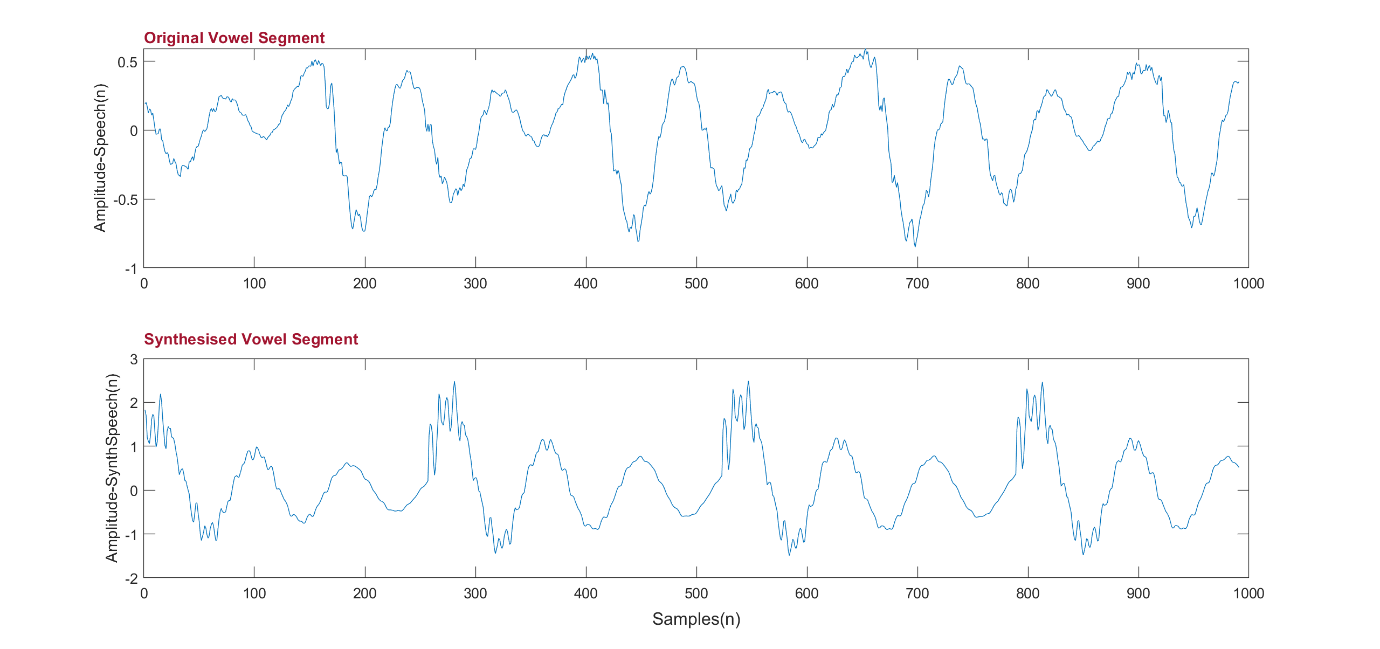


Figure 5: Synthesis of heed\_m

# 3 Analysis

## 3.1 LPC Order Variation

The effect of varying the order of LPC filter coefficients was observed on the LPC filter response and the synthesised vowel segment.

The effect on filter response can be visually inspected in figure 2, figure 6 and figure 7. An interesting observation was made by comparing the frequency response of male and female vowel. In general, the male’s formant frequencies were lower than the female’s but here, the filter did not accurately map the high magnitude of low frequencies for female sample. Hence, the first formant frequency was lower than expected. This effect can be mitigated partly by utilising higher order numbers.

For both voice samples at lower orders, the filter’s response did not map well to the peaks and valleys in the PSD and appeared to be smooth. As the order was increased, the frequency response mapped the variations in PSD more accurately and at very high orders it followed all the major variations.

The effect on synthesised vowel segment can be visually inspected in figure 4, figure 5, figure 8 and figure 9.

In the amplitude plots it was observed that as the order increases the synthesised vowel segment estimated the original vowel segment more accurately.

The spectrogram plots illustrated that higher frequencies are more profound in the female vowel segment. At lower orders the individual frequencies blurred together and appeared to have a lower resolution. Thus, the formant frequencies could not be deduced. As the order was increased, the individual frequencies became clearer and formant frequencies could be estimated. It must also be noted that the refinement in resolution from order 30 to 90 wasn’t as profound as it was when going from 1 to 30.

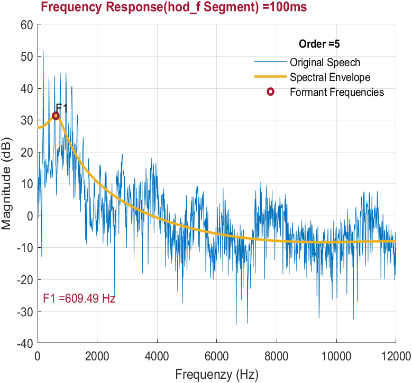
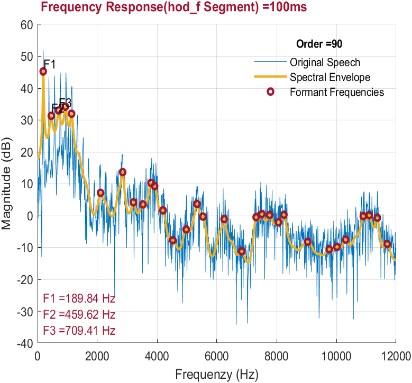


Figure 6: Effect of LPC order on PSD of hod\_f

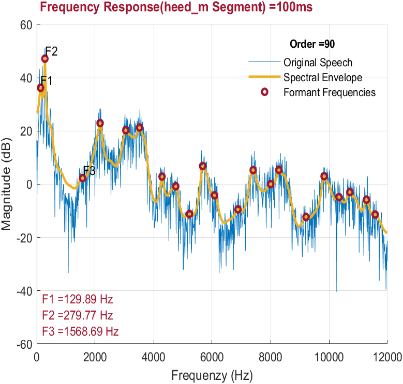
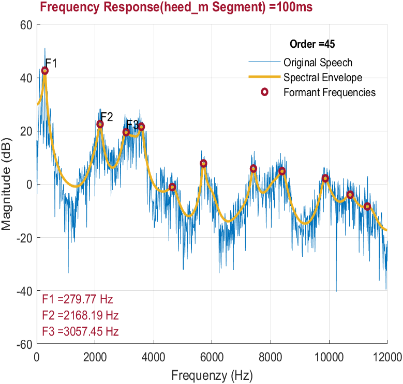
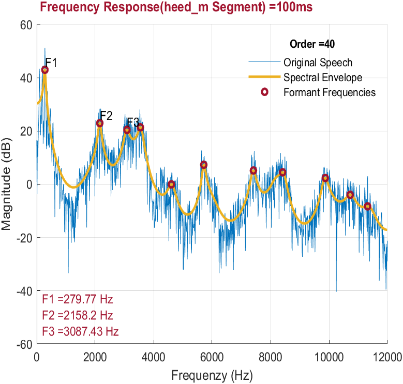
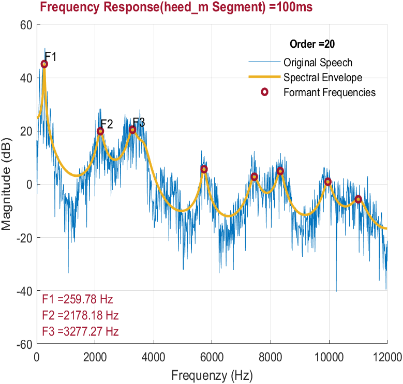
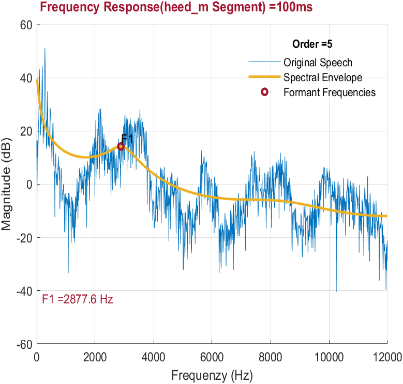
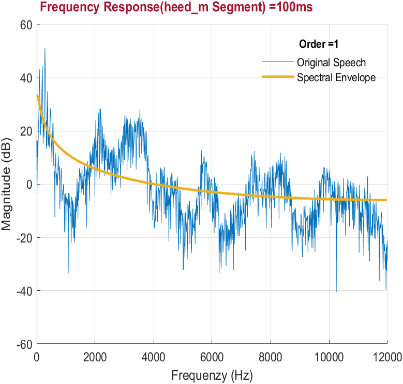


Figure 7: Effect of LPC order on PSD of heed\_m

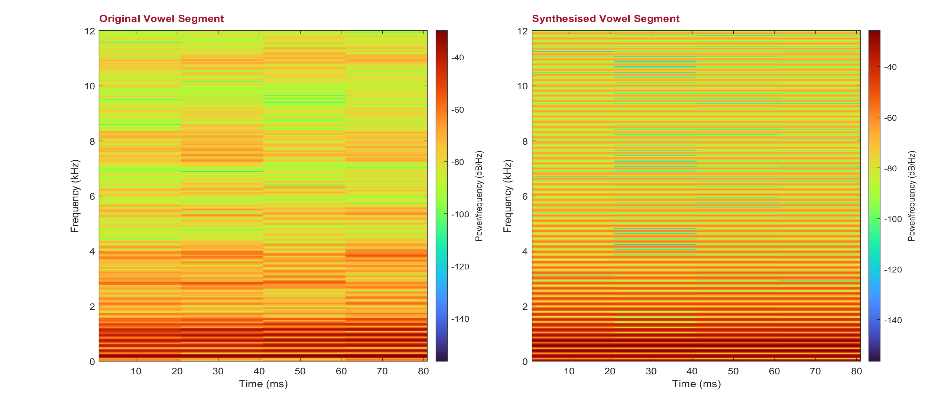
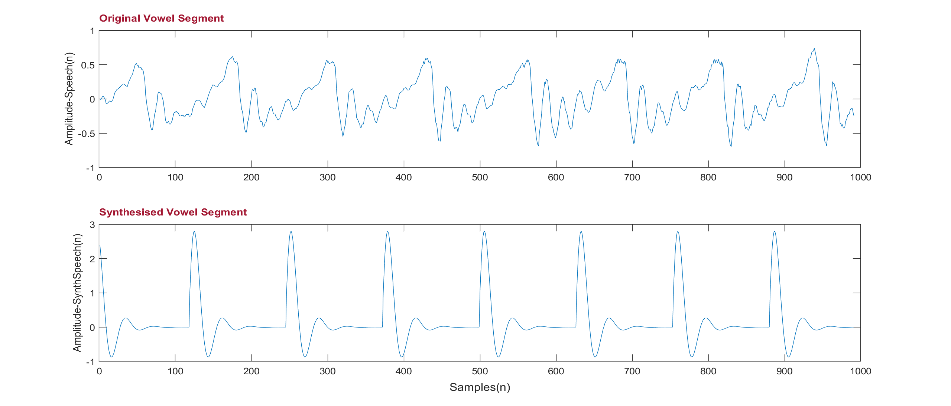


Figure 8: Synthesis of hod\_f - LPC order = 5

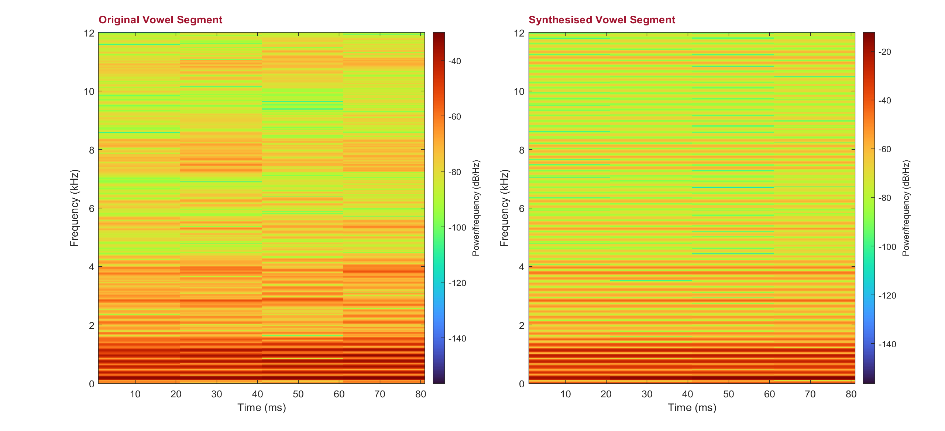
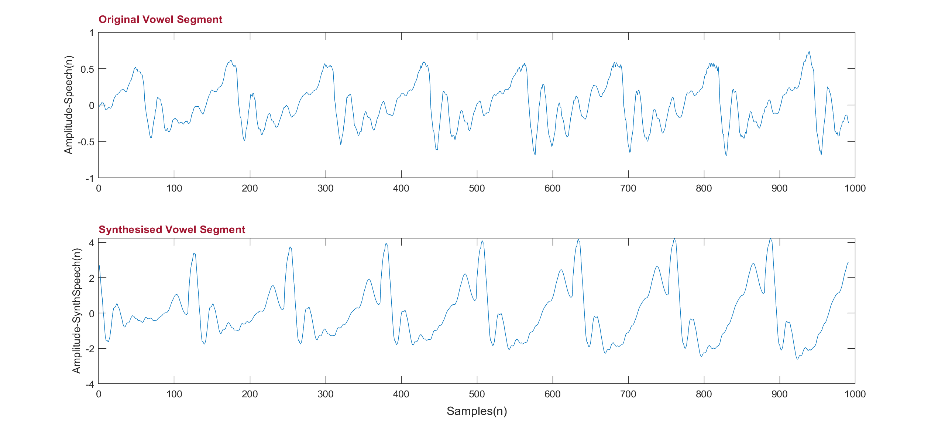


Figure 9: Synthesis of hod\_f - LPC order = 90

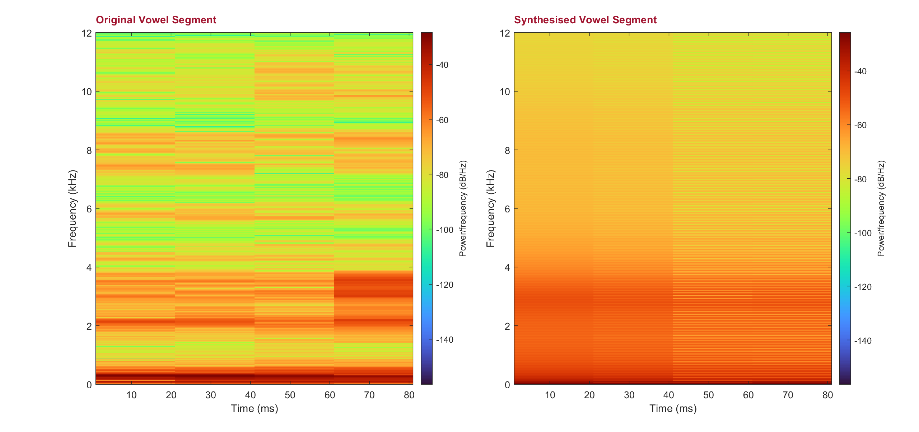
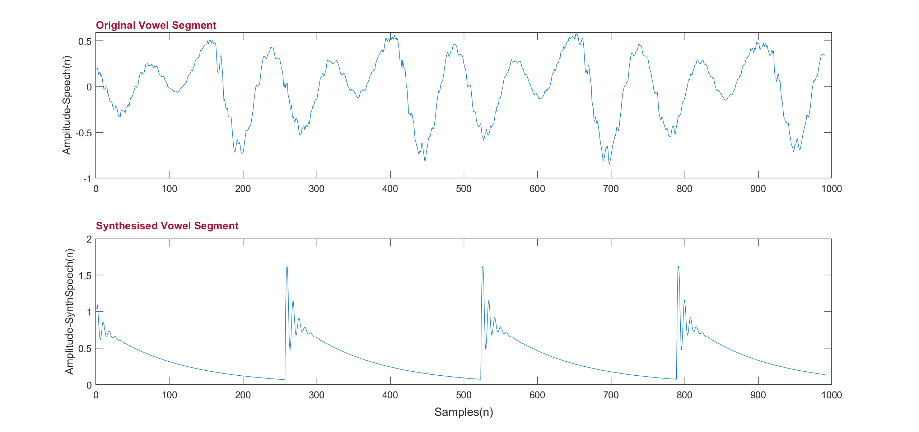


Figure 10: Synthesis of heed\_m - LPC order = 5

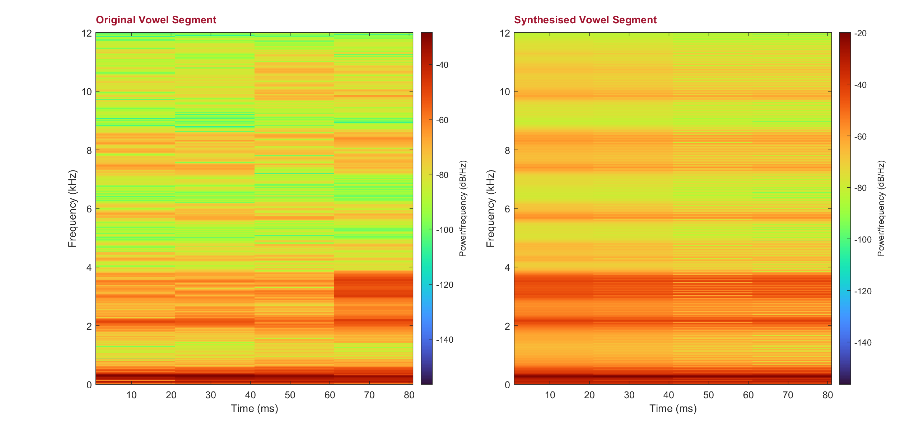
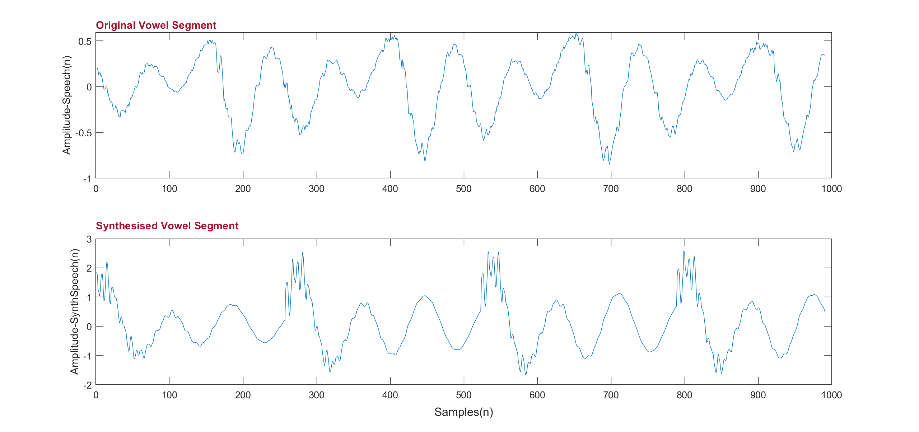


Figure 11: Synthesis of heed\_m - LPC order = 90

## 3.2 Source Segment Length Variation

The variation of source segment length has an interesting effect on the LPC filter response. As the source segment length was reduced, the spectral envelop became smoother and the extent of the peaks and valleys was reduced. The first few formants have a greater tolerance to this variation compared to the higher frequencies. This can be observed in figure 10.

The effect of source segment length variation can also be observed on the spectrogram of the original and synthesised vowel segments. As the length was reduced, the frequencies seemed to spread out and it became difficult to pinpoint the exact frequency. All this was done while keeping width of window constant. This effect was less profound on the lower frequencies. This can be observed in figure 11.

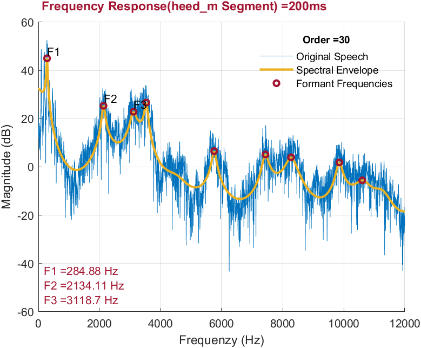
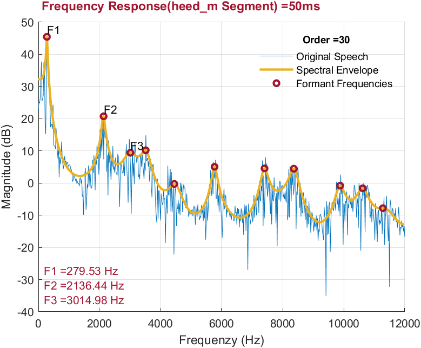
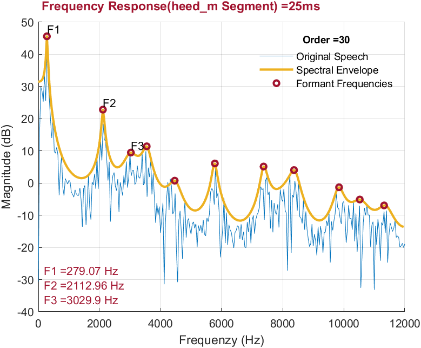


Figure 12: Effect of Segment Length on PSD of heed\_m

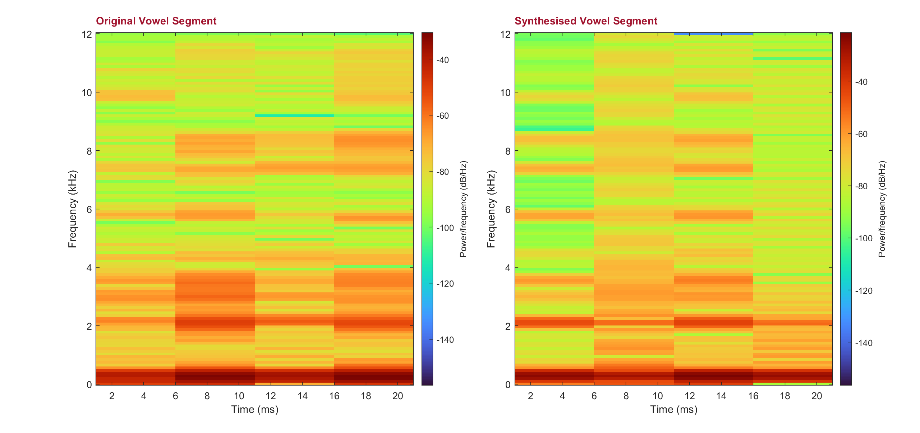


Figure 13: Synthesis of heed\_m - Segment Length = 25

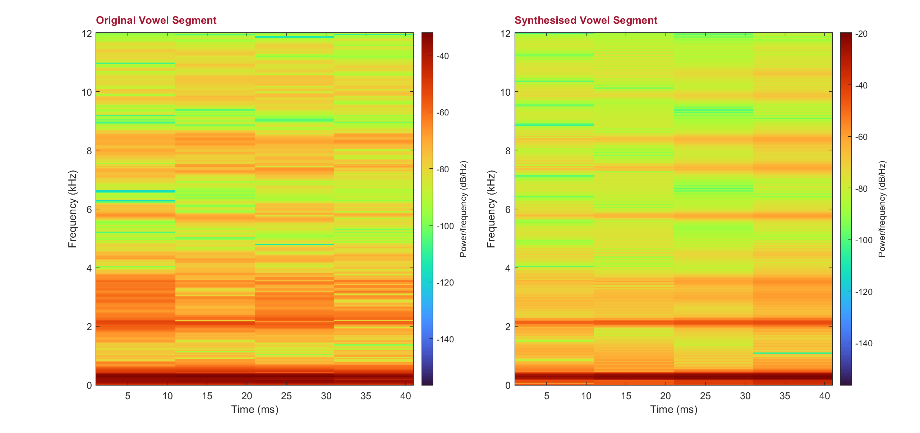


Figure 14: Synthesis of heed\_m - Segment Length = 50

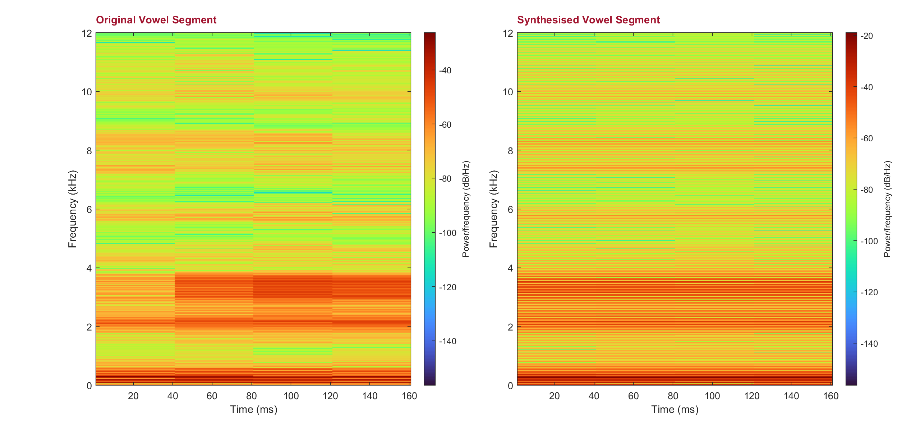


Figure 15: Synthesis of heed\_m - Segment Length = 200

## 3.3 Auditory Analysis

At lower orders of LPC coefficients, the synthesised vowel segment sounds like an electronic buzzer and cannot be identified as speech. When the order was increased to about 20, the synthesised segment resembled speech. The sound was still harsh, but the actual vowel was somewhat discernible. At the same value for order the synthesised male voice was much clearer than the female voice. When the order was increased to about 35, the synthesised voice was very clear, especially for male voice and increasing the order further lead to a smoother voice but did not reach the quality of the original voice.

Altering the length of source segment also influenced the synthesised sound segment. While keeping order constant, the segment length was increased, and it was observed that the quality of audio signal is poor at low segment lengths, but the vowel is still discernible. Going beyond 100ms in segment lead to no subjective increase in quality of synthesised voice.

# 4 Conclusion

This report presents the development and analysis of a Linear Predictive Speech Synthesizer. The speech synthesizer was based on the source-filter model of speech. This involved the analysis of vowel samples and calculation of formant frequencies and fundamental frequency. Speech segments were synthesised comparable to human speech. Finally, the relationship of quality of synthesised speech to system variables such as LPC filter order and source segment length was also investigated.

# References

1. Čuperlović-culf, m., 2013. 4 - metabolomics nmr data preprocessing – analysis of individual spectrum. In: m. Čuperlović-culf, ed. *Nmr metabolomics in cancer research.* S.l.:woodhead publishing, pp. 215-259.
2. Deng, l. & o'shaughnessy, d., 2018. *Speech processing: a dynamic and optimization-oriented approach.* S.l.:crc press.
3. Mannell, r., 2020. *Source-filter theory.* [online]   
   available at: https://www.mq.edu.au/about/about-the-university/our-faculties/medicine-and-health-sciences/departments-and-centres/department-of-linguistics/our-research/phonetics-and-phonology/speech/acoustics/acoustic-theory-of-speech-production/source-filter-theory  
   [accessed 1 november 2021].
4. Oppenheim, a. V. & schafer, r., 2004. From frequency to quefrency: a history of the cepstrum. *Signal processing magazine, ieee,* october, volume 21, pp. 95-106.
5. Smith, j. O., 2007. *Introduction to digital filters: with audio applications.* S.l.:w3k.
6. Yung-li, l., 2005. 10 - fatigue analysis in the frequency domain. In: l. E. E. Yung-li, p. A. N. Jwo, r. B. Hathaway & m. E. Barkey, eds. *Fatigue testing and analysis.* Burlington: butterworth-heinemann, pp. 369-394.

# 6 Code

%% Linear Predictive Speech Synthesizer

% \_EEEM030 Assignment 1\_

%% Function List

%

% \* Offset\_func.m

% \* Segment\_func.m

% \* FFT\_func.m

% \* LPC\_func.m

% \* Cepstrum\_func.m

% \* Synthesis\_func.m

%% Constants

%close all; clear all ;clc;

speech = 'hod\_f'; % file name

[original\_speech\_t,sampling\_freq]=audioread(strcat(speech,'.wav'));

speech\_t = original\_speech\_t;

segment\_t = 100; % time segment in ms

offset\_t = 20; % offset in ms

lpc\_order = 30; % no. of LPC coefficients

annotation\_str = ""; % annotation strings

formants\_i = 3; % display first 3 formant frequencies

cepstrum\_threshold = 0.05; % cutoff threshold for cepstrum

quefrency\_threshold = 50; % cutoff threshold for quefrency

synthesised\_speech\_t = 1000; % synthesised speech length in ms

w\_number = 5; % number of windows for spectrogram

w\_overlap = 2; % overlap in ms

synthesised\_w\_number = 5; % number of windows for spectrogram

synthesised\_w\_overlap = 2; % overlap in ms

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Pre-processing

%%

% \*Offset & Segment\*

speech\_t = Offset\_func(speech\_t,offset\_t,sampling\_freq);

speech\_t = Segment\_func(speech\_t,segment\_t,sampling\_freq);

figure(1) % plot speech segment

hold on

plot(speech\_t,'Color',[0 0.4470 0.7410])

t = title(speech,'Interpreter','none');

ax = gca;

ax.TitleHorizontalAlignment = 'left';

t.Color = [0.6350 0.0780 0.1840];

xlabel('Samples(n)')

ylabel('Amplitude-Speech(n)')

hold off

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Model Estimation

%%

% \*FFT\*

speech\_fft = FFT\_func(speech\_t,length(speech\_t));

figure (2) % plot frequency response for speech signal and LPC filter

hold on

freq\_scale = sampling\_freq\*(0:(length(speech\_t)/2))/length(speech\_t);

orig\_freq\_plot = plot(freq\_scale,20\*log10(abs(speech\_fft)),'Color',[0 0.4470 0.7410]);

orig\_freq\_plot.LineWidth = 0.5;

t = title({'PSD of'; speech},'Interpreter','none');

ax = gca;

ax.TitleHorizontalAlignment = 'left';

t.Color = [0.6350 0.0780 0.1840];

xlabel('Frequenzy (Hz)')

ylabel('Magnitude (dB)')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%

% \*LPC & Freq Response\*

[lpc\_coeff, freq\_res\_values\_db, freq\_res\_freqs, peaks\_freqs, peaks] = LPC\_func(speech\_t,lpc\_order,freq\_scale,sampling\_freq);

lpc\_freq\_plot = plot(freq\_res\_freqs, freq\_res\_values\_db);

lpc\_freq\_plot.LineWidth = 2;

lpc\_freq\_plot.Color = [0.9290 0.6940 0.1250];

peaks\_plot = plot(peaks\_freqs, peaks, 'o','Color',[0.6350 0.0780 0.1840]); %Display first 3 formant frequencies

for i = 1:formants\_i

text(peaks\_freqs(i),peaks(i),{strcat('F',num2str(i))},'VerticalAlignment','bottom','FontSize',10);

str = (strcat('F',num2str(i),' = ',num2str(round(peaks\_freqs(i),2)),' Hz'));

annotation\_str = cat(2,annotation\_str,str);

end

annotation\_str(cellfun('isempty',annotation\_str)) = []; % remove empty cells from annotation\_str

a = annotation('textbox','String',annotation\_str,'EdgeColor','none','Color',[0.6350 0.0780 0.1840]);

a.Position = [0.13,0.155,0.1,0.1]; %annotation position

peaks\_plot.MarkerSize = 5;

peaks\_plot.LineWidth = 2;

grid

lgd = legend('Original Speech', 'Spectral Envelope', 'Formant Frequencies');

legend('boxoff');

lgd.Title.String = strcat('Order = ',num2str(lpc\_order)) ;

t = title(strcat(' Frequency Response ','(', speech,' Segment) = ',num2str(segment\_t),'ms'));

t.Color = [0.6350 0.0780 0.1840];

ax = gca;

ax.TitleHorizontalAlignment = 'left';

hold off

figure(3) % plot pole zero plot

zplane(1,lpc\_coeff);

t = title('Pole-Zero Plot');

t.Color = [0.6350 0.0780 0.1840];

ax = gca;

ax.TitleHorizontalAlignment = 'left';

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%

% \*Cepstrum & Fundamental Frequency\*

fundamental\_freq = Cepstrum\_func(speech,speech\_t, cepstrum\_threshold, quefrency\_threshold, sampling\_freq);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Synthesis

%%

% \*Impulse Train & Synthesis\*

synthesised\_speech = Synthesis\_func(lpc\_coeff, fundamental\_freq, sampling\_freq, synthesised\_speech\_t);

figure(6) % plot original and synthesised vowel segments

hold on

synthesised\_speech\_plot = synthesised\_speech(10:1000);

speech\_plot = speech\_t(10:1000);

t = tiledlayout(2,1);

xlabel(t,'Samples(n)')

ax1 = nexttile; % Top (plot of Original Vowel Segment)

plot(ax1,speech\_plot,'Color',[0 0.4470 0.7410])

ylabel(ax1,'Amplitude-Speech(n)')

t1 = title(ax1,'Original Vowel Segment');

t1.Color = [0.6350 0.0780 0.1840];

ax1.TitleHorizontalAlignment = 'left';

ax2 = nexttile; % Bottom (plot of Synthesised Vowel Segment)

plot(ax2,synthesised\_speech\_plot,'Color',[0 0.4470 0.7410])

ylabel(ax2,'Amplitude-SynthSpeech(n)')

t2 = title(ax2,'Synthesised Vowel Segment');

t2.Color = [0.6350 0.0780 0.1840];

ax2.TitleHorizontalAlignment = 'left';

hold off

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%

% \*Spectrogram\*

figure(7) % plot spectrogram of original and synthesised vowel segments

tiledlayout(1,2)

sample\_overlap = round((w\_overlap / 1000) \* sampling\_freq); % Original Vowel Spectrogram

window\_size = round((length(speech\_t) + (w\_number) \* sample\_overlap)/(w\_number));

ax1 = nexttile;

spectrogram(speech\_t,window\_size,sample\_overlap, [], sampling\_freq,'yaxis');

t1 = title(ax1,'Original Vowel Segment');

t1.Color = [0.6350 0.0780 0.1840];

ax1.TitleHorizontalAlignment = 'left';

ylabel(ax1,'Frequency (kHz)')

colormap turbo

sample\_overlap = round((synthesised\_w\_overlap / 1000) \* sampling\_freq); % Synthesised Vowel Spectrogram

window\_size = round((length(synthesised\_speech) + (synthesised\_w\_number) \* sample\_overlap)/(synthesised\_w\_number));

ax2 = nexttile;

spectrogram(synthesised\_speech,window\_size,sample\_overlap, [], sampling\_freq,'yaxis');

t2 = title(ax2,'Synthesised Vowel Segment');

t2.Color = [0.6350 0.0780 0.1840];

ax2.TitleHorizontalAlignment = 'left';

ylabel(ax2,'Frequency (kHz)')

colormap turbo

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%

% \*Audio Play & Write\*

ap1 = audioplayer(synthesised\_speech,sampling\_freq,16);

ap2 = audioplayer(original\_speech\_t,sampling\_freq,16);

play(ap2);

pause(2);

play(ap1);

%audiowrite(strcat('synthesised\_',speech,'\_',num2str(segment\_t),'\_',num2str(lpc\_order),'.wav'),synthesised\_speech,sampling\_freq);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Offset

function offset\_speech\_t = Offset\_func(speech\_t,offset\_t,sampling\_freq)

offset\_n = max((offset\_t / 1000) \* sampling\_freq,1) ;

offset\_speech\_t = speech\_t(offset\_n : end);

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Segment

function speech\_segment = Segment\_func(speech\_t, segment\_t, sampling\_freq)

segment\_n = (segment\_t / 1000) \* sampling\_freq ;

segment\_n = min(segment\_n, length(speech\_t));

speech\_segment = speech\_t(1 : segment\_n);

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% FFT

function speech\_fft = FFT\_func(speech\_t,speech\_length)

speech\_fft = fft(speech\_t);

speech\_fft = abs(speech\_fft);

speech\_fft = speech\_fft(1:floor(speech\_length/2+1));

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% LPC & Frequency Response

function [lpc\_coeff, freq\_res\_values\_db, freq\_res\_freqs, peaks\_freqs, peaks] = LPC\_func(speech\_t,lpc\_order,freq\_scale,sampling\_freq)

lpc\_coeff = lpc(speech\_t,lpc\_order);

[freq\_res\_values, freq\_res\_freqs] = freqz(1 , lpc\_coeff , length(freq\_scale), sampling\_freq);

freq\_res\_values\_db = 20\*log10(abs(freq\_res\_values));

peaks = islocalmax(freq\_res\_values\_db);

peaks\_freqs = freq\_res\_freqs(peaks);

peaks = freq\_res\_values\_db(peaks);

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Cepstrum & Fundamental frequency

function fundamental\_freq = Cepstrum\_func(speech,speech\_t,cepstrum\_mag\_threshold,quefrency\_threshold,sampling\_freq)

cepstrum = rceps(speech\_t);

cep\_quefrency =(1:length(speech\_t));

cepstrum = cepstrum(1:round(length(speech\_t) / 2));

cep\_quefrency = cep\_quefrency(1:round(length(speech\_t) / 2));

figure(4) % plot cepstrum & pitch period

cepstrum\_plot = plot(cep\_quefrency,cepstrum,'Color',[0 0.4470 0.7410]);

cepstrum\_plot.LineWidth = 0.5;

grid

xlabel('Quefrency')

ylabel('Real Cepstrum(x[n])')

xlim([0 length(speech\_t) / 2])

t = title(strcat('Speech Cepstrum ','(',speech,')'),'Interpreter','none');

t.Color = [0.6350 0.0780 0.1840];

ax = gca;

ax.TitleHorizontalAlignment = 'left';

hold on

cep\_que\_thresh = quefrency\_threshold < cep\_quefrency; % Cut below quefrency threshold

cep\_quefrency = cep\_quefrency(cep\_que\_thresh);

cepstrum = cepstrum(cep\_que\_thresh);

cep\_mag = islocalmax(cepstrum); % Find local maxima

cep\_quefrency = cep\_quefrency(cep\_mag);

cepstrum = cepstrum(cep\_mag);

cepstrum(cepstrum < cepstrum\_mag\_threshold) = 0; % Cut below cepstrum value threshold

pitch\_period = cep\_quefrency(cepstrum == max(cepstrum));

fundamental\_freq = 1 / (pitch\_period / sampling\_freq);

maxima\_plot = plot(pitch\_period, cepstrum(cepstrum == max(cepstrum)), 'o');

maxima\_plot.MarkerSize = 5;

maxima\_plot.LineWidth = 2;

text(cep\_quefrency(cepstrum == max(cepstrum)),cepstrum(cepstrum==max(cepstrum)),{strcat('Pitch Period =',num2str(pitch\_period))},'VerticalAlignment','bottom');

annotation\_str = {'F\_0 = 1 \div P\_0/F\_S' ,(strcat('F\_0 = ',num2str(round(fundamental\_freq,2)),' Hz'))};

a = annotation('textbox','String',annotation\_str,'Color',[0.6350 0.0780 0.1840]);

a.Position = [0.755,0.81,0.1,0.1];

a.FontWeight = 'bold';

a.BackgroundColor = [0.9290 0.6940 0.1250];

a.FaceAlpha = 0.7;

a.HorizontalAlignment = 'center' ;

a.VerticalAlignment = 'middle' ;

hold off

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Synthesis

function synthesised\_speech = Synthesis\_func(lpc\_coeff, fundamental\_freq, sampling\_freq, synthesised\_speech\_t)

required\_samples = (synthesised\_speech\_t / 1000) \* sampling\_freq;

impulse\_train = zeros(1,required\_samples);

impulse\_train(1:round(sampling\_freq/fundamental\_freq):end)=1;

synthesised\_speech = filter(1, lpc\_coeff, impulse\_train);

figure(5) % plot impulse train

hold on

plot(impulse\_train,'Color',[0 0.4470 0.7410]);

t = title('Impulse Train');

ax = gca;

ax.TitleHorizontalAlignment = 'left';

t.Color = [0.6350 0.0780 0.1840];

xlabel('Samples(n)')

ylabel('Amplitude-Fx(n)')

hold off

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%