Introductory Speech Processing Workshop

Instructor: Ciira wa Maina
Dedan Kimathi University of Technology,
Department of Electrical and Electronic Engineering,
P.O. BOX 657-10100 Nyeri, Kenya
E-mail: ciira.maina@dkut.ac.ke

Objectives

The main objectives of this laboratory are to:

1. Introduce the student to the idea of fundamental frequency.
2. Demonstrate the estimation of fundamental frequency from sampled speech.
3. Introduce the student to the idea of voiced and unvoiced speech.
4. Introduce the student to speech modelling using linear prediction coefficients.

Theoretical Background

Speech Processing

Human speech is arguably one of the most important signals encountered in engineering applications. Numerous devices record and manipulate speech signals to achieve different ends. To properly manipulate the signal, it is important to have an understanding of the speech production process. The lungs, vocal tract and vocal cords all play an important role in speech production [1]. The speech production model consists of an input signal from the lungs and a linear filter. In this model, the input is a white noise process which is spectrally flat. This input is then spectrally shaped by a filter which models the properties of the vocal tract. Since the properties of the vocal tract are constantly changing as different sounds are produced, the filter is time varying. However, the filter is often modelled as quasi-stationary with filter parameters constant over a period of approximately 30ms.

When the vocal cords vibrate as is the case when pronouncing the sound /a/ in cat, we say that the sound is voiced and in this case the signal is seen to be exhibit some periodicity (see Figure 5(a)). When the vocal cords do not vibrate the sound is unvoiced.

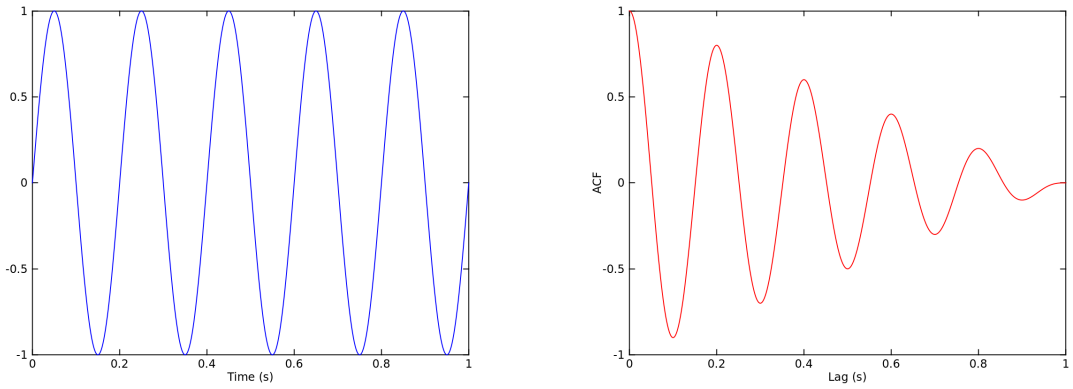
Estimation of Fundamental Frequency

When speech is voiced, it is seen to exhibit periodicity and it is often important in speech applications to estimate the pitch of these signals. To achieve this, we estimate the *fundamental frequency* of this signal also refered to as $F0$. A popular method for estimation of $F0$ is based on the autocorrelation function (ACF).

Consider a periodic signal $\cos(2\pi f_0 t)$ which oscillates at a frequency f_0. To work with this signal on a computer we sample it at a frequency $f_s = \frac{1}{T_s}$ to form a discrete time signal $x[n] = \cos(2\pi f_0 n T_s)$. We can compute the ACF of the signal $x[n]$ using

$$R_x[k] = \frac{1}{N} \sum_{n=0}^{N-k-1} x[n]x[n+k] \quad 0 \leq k \leq N-1 \tag{1}$$

where k is the time lag and N is the length of the signal in samples. This function measures the similarity between samples at particular times and those obtained at particular time lags. If the signal is periodic we expect this function to have peaks at lags equivalent to integer multiples of the signal period in addition to a peak at zero lag.

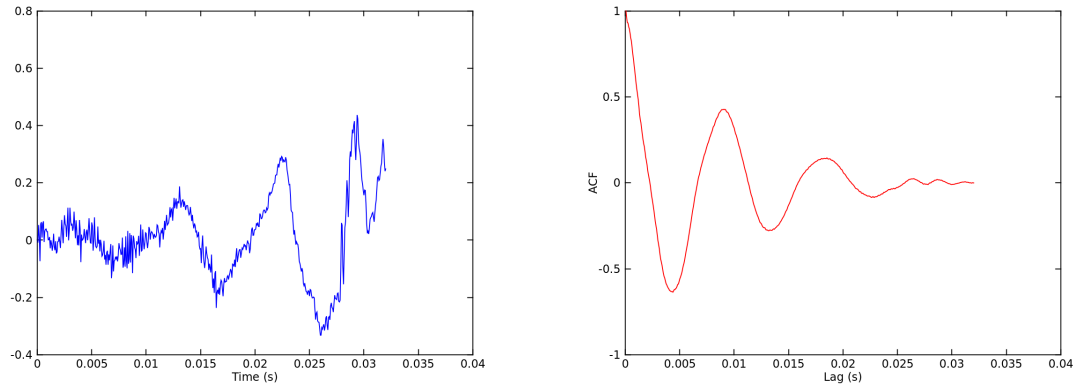


(a) (b)

Figure 1. A periodic sine wave (a) and the corresponding autocorrelation function (b).

If we form a finite duration signal from $x[n]$ by considering the signal over a finite interval and we compute the ACF, we notice that it has peaks at lags corresponding to integer multiples of the period as shown in Figures 1(a) and 1(b) .

To apply this method to a speech signal, we compute the ACF of a finite duration signal corresponding to a speech segment 32ms long. Over this short segment the characteristics of the signal can be assumed to be stationary. Figures 2(a) and 2(b) show the speech signal and ACF of a speech signal obtained from the author vocalising the word 'moja' which means 'one' in Kiswahili. From the plot of the ACF we see that the first non zero peak is obtained at aproximately $0.01s$ which corresponds to an estimate of $F0 = 100$Hz.



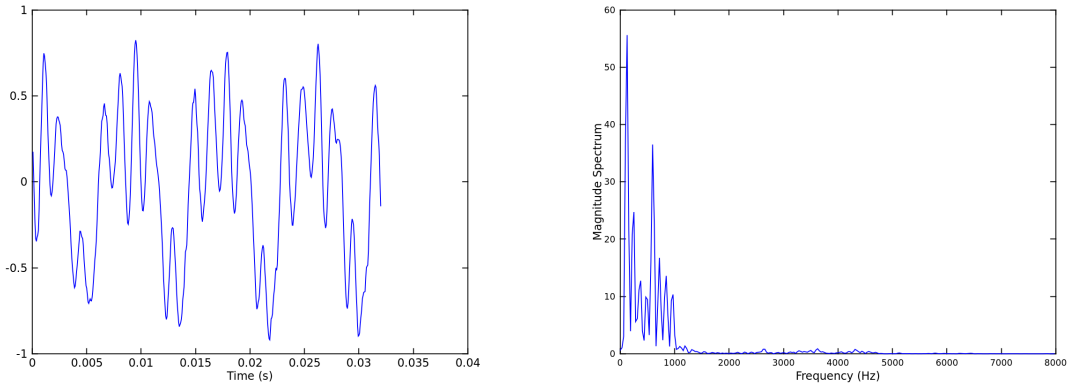
(a) (b)

Figure 2. Speech waveform of a voiced speech segment (a) and the corresponding autocorrelation function (b).

Measuring Frequency Content of a Signal: The Spectrogram

Several applications require the analysis of frequency content of signals. In continuous time signals this is achieved using the Fourier series (for periodic signals) or the Fourier transform (for aperiodic signals). In real world aplications continuous time signals are first discretized before they are analysed. In this case we use the Discrete Time Fourier Transform (DTFT) to analyze the signals. The DTFT is a sampled version of the frequency spectrum of the sampled signal and it is efficiently implemented using the Fast Fourier Transform (FFT).

To compute the frequency content of the speech signals we divide the signal into overlapping segments each 32ms long. We then compute the DTFT of the segments and obtain a sampled version of the spectrum.[1] This produces a 2-dimesional image known as a *spectrogram*. Figure 4 shows a spectrogram obtained from a recording of the author saying 'moja'. It is obtained from a 2s recording sampled at 16kHz. It appears as a heatmap with high values represented by deep red and low values deep blue. Intermediate values appear yellow. From this spectrogram we see that the word is spoken in the interval between $0.8s$ and $1.4s$. If we extract the segment at 1s corresponding to 32ms (512 samples) and compute it's DTFT we obtain the results shown in Figure 3. We see that most of the spectral content is below 1 kHz.



(a) (b)

Figure 3. Speech waveform of a voiced speech segment (a) and the corresponding magnitude spectrum (b).

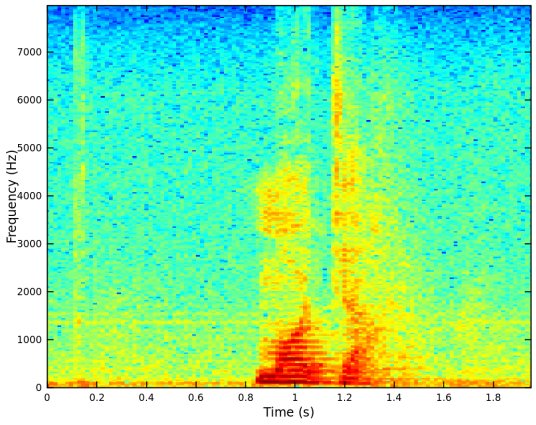


Figure 4. Spectrogram of the word 'moja'.

Linear prediction coefficients

Linear prediction coefficients (LPCs) provide a good and analytically tractable model for speech when the sound is voiced [2]. The idea behind LPCs is that a given speech sample s_n can be accurately approximated

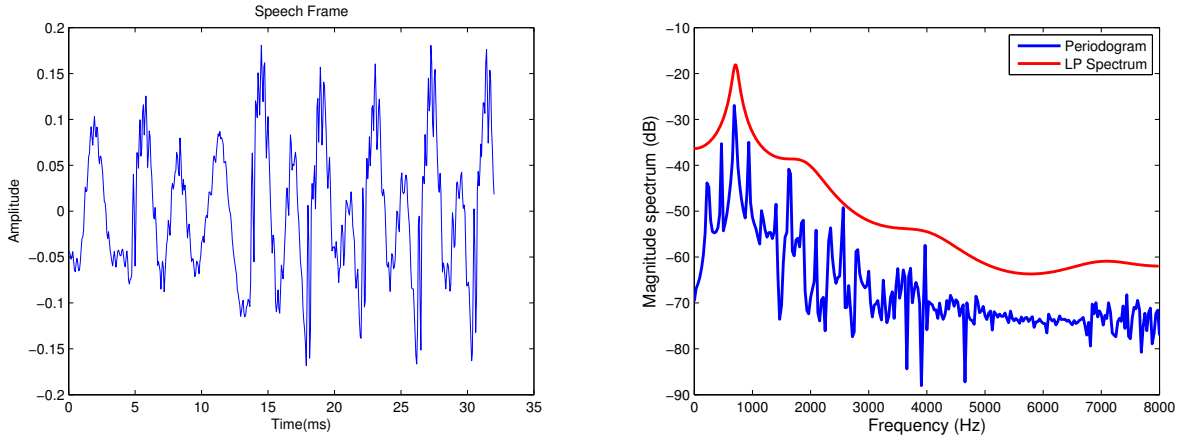
[1] In general this is a complex function and we must plot both the magnitude and phase for a complete representation. In speech applications we often ignore phase and work with the magnitude spectrum.

using a linear combination of P previous samples. That is

$$s_n \approx -\sum_{i=1}^{P} a_i s_{n-i}. \tag{2}$$

The coefficients $a_1, \ldots, a_P$ are constant for a given speech frame.

Determining the LPCs corresponding to a speech segment is equivalent to determining the parameters of the filter that spectrally shapes a white noise input to produce the sound. A plot of the frequency response of this filter reveals the frequency spectrum of the sound and in the case of voiced speech exhibits spectral peaks. These peaks are known as formants. Figure 5(b) shows the frequency spectrum of the voiced segment shown in Figure 5(a). The spectrum is estimated using both the periodogram (magnitude of the DTFT) and the LPCs. When the LPCs are used the spectrum estimate is less noisy and spectral peaks can be clearly identified. The location of the lowest frequency peak provides an estimate for the first resonance peak or formant denoted $F1$. Successive formants are denoted $F2$, $F3$ etc.



(a) (b)

Figure 5. Speech waveform of a voiced speech segment (a) and Typical Linear Prediction Spectrum (b).

Procedure

This lab will be performed on a computer running a linux operating system (preferably Ubuntu) with necessary hardware accessories and necessary software installed. In order to perform the lab you will need a microphone and a pair of headphones. Also, the following software will need to be installed

- Octave- A high level language suitable for numerical computations that is quite similar to Matlab. We will also require the signals package.
- SoX - Sound eXchange, the Swiss Army knife of audio manipulation

Frequency Estimation

1. Generate a sinusoidal signal over the interval $0 \leq t \leq 1$ oscillating at a frequency of 5 Hz.

   ```
   t=0:0.001:1;
   x=sin(2*pi*5*t);
   ```

2. Plot this function

   ```
   figure(1)
   clf()
   ```

```
plot(t,x,"linewidth", 2)
xlabel('Time (s)', "fontsize", 18)
```

3. Compute the autocorrelation function and plot it

```
rx=autocor(x);
figure(2)
clf()
plot(t,rx,'r',"linewidth", 2)
xlabel('Lag (s)', "fontsize", 18)
ylabel('ACF', "fontsize", 18)
```

4. Determine the location of the first peak of the ACF after the peak at $t = 0$. The estimated value of the frequency of oscillation is the reciprocal of this value.

Record Some Speech

1. Using the rec function of the SoX package create a two second recordings of yourself saying one of the 10 digits in Kiswahili.

   (a) Start the recording for two seconds

   ```
   rec -c 1 -r 8000 digit.wav trim 0 2
   ```

   This will record a single channel audio signal for 2 seconds sampled at 8 kHz and store it in digit.wav

   (b) Say a digit

   (c) Play the file to ensure you actually recorded yourself. If you are not satisfied with the recording, try again

   ```
   play test.wav
   ```

Compute a spectrogram

We will now compute a spectrogram of the speech signal recorded in the previous section.

1. Launch Octave.

2. Load the speech signal recorded in the previous section.

   ```
   [x, Fs, bps] = wavread ('test.wav' );
   ```

3. We will compute the spectrogram by dividing the speech signal into segments 32ms long (256 samples) with 50% ovelap between neighboring segments and taking the FFT of each segment. We plot the magnitude of this FFT for all the segments as a 2-dimensional image using the function specgram.

```
alpha=0.5; %Overlap
N=256;% 32ms window size
figure(2)
clf()
specgram(x,N,Fs,[],alpha*N);
xlabel('Time (s)', "fontsize", 18)
ylabel('Frequency (Hz)', "fontsize", 18)
```

   (a) Can you tell from the spectrogram the points in the recording containing the spoken digit?

   (b) Change N so that the segments are 64 ms long and plot a new spectrogram. What is the effect on the spectrogram?

   (c) In what frequency range is most of the energy of your signal?

   (d) Can you identify clear peaks in the magnitude spectrum?

$F0$ Estimation

We will now estimate the fundamental frequency of a speech segment using the autocorrelation method.

1. From your spectrogram, determine a time point t in the speech signal with distinct peaks in the magnitude spectrum.
2. Determine the segment number using the relation

$$seg\_num = \frac{t - \frac{window\_size}{2}}{(1-\alpha) \times window\_size}$$

   where t is the time point of interest. $window\_size$ is the length of segment in units of time (same units as t) and α is the percentage overlap expressed as a fraction. For example if we are interested in the segment at $t = 1s$ we get

$$seg\_num = \frac{1 - \frac{0.032}{2}}{(1-0.5) \times 0.032}$$

   Since seg_num must be an integer we round off the result to the nearest integer.
3. Extract the segment from the rest of the recording

```
y=x((seg_num-1)*(1-alpha)*N+1:(seg_num-1)*(1-alpha)*N+N);
```

4. Compute the autocorrelation function of the segment and plot the segment and ACF. To plot the signal and ACF as a function of time we multiply the sample number by the sampling period

```
ry=autocor(y);
t1=1:length(y);
t2=t1*(1/Fs);
figure(3)
clf()
plot(t2,y,"linewidth", 2)
xlabel('Time (s)', "fontsize", 18)
set(gca, "linewidth", 2, "fontsize", 18)
figure(4)
clf()
plot(t2,ry,'r',"linewidth", 2)
xlabel('Lag (s)', "fontsize", 18)
ylabel('ACF', "fontsize", 18)
set(gca, "linewidth", 2, "fontsize", 18)
```

5. Now like we did when estimating the frequency of the sinusoid, determine the location of the first peak of the ACF after the peak at $t = 0$. The estimated value of $F0$ is the reciprocal of this value.
6. Compare your value of $F0$ with colleagues. Do you notice any differences with colleagues of the opposite gender?

Linear Prediction Coefficients

In the final section of this laboratory, we will estimate the linear prediction coefficients of the speech segment and estimate the transfer function of the vocal tract. The idea behind LPCs is that a given speech sample s_n can be accurately approximated using a linear combination of P previous samples. That is

$$s_n \approx -\sum_{i=1}^{P} a_i s_{n-i}.$$

The coefficients $a_1, \ldots, a_P$ are constant for a given speech frame. It can be shown that the transfer function of the vocal tract when producing the segment is

$$H(z) = \frac{G}{1 + \sum_{i=1}^{P} a_i z^i}$$

where G is a gain term. The formant frequencies are derived from the zeros of the denominator of $H(z)$.

The LPCs are computed using the Levinson Durbin algorithm which derives the LPCs from the auto-correlation function.

1. Set the LPC order $P = 8$

   ```
   p=8;%order
   ```

2. Compute the ACF up to P lags

   ```
   acf=autocor(y,p);
   ```

3. Compute the LPCs using the Levinson Durbin algorithm

   ```
   [a,G2]=levinson(acf);
   ```

   The function also computes G^2 which is the square of the gain term in $H(z)$

4. Compute and plot the frequency response of the vocal tract filter

   ```
   [h,w]=freqz(sqrt(G2),a,N,Fs);
   figure(5)
   clf()
   plot(w,20*log10(abs(h)))
   ```

5. We can also compare the spectrum we obtain from the LPCs with that obtained by computing the FFT of the signal. We will use the specgram function to compute the spectrogam and return the FFTs of all segments and the extract that corresponding to the segment we are dealing with.

   (a) Compute the FFTs of all segments

   ```
   [S,f,t]=specgram(x,N,Fs,[],alpha*N);
   ```

   (b) Plot the magnitude spectrum of the speech segment and the LPC spectrum on the same plot

   ```
   figure(6)
   clf()
   plot(f,20*log10(abs(S(:,seg_num))))
   hold on
   plot(w,20*log10(abs(h)),'-r')
   ```

6. How do the LPC spectrum and magnitude spectrum compare?

7. Does the LPC spectrum exhibit clear peaks?

8. Try different LPC orders such as 16, 32 etc and see if anything changes.

References

1. Campbell Jr JP (1997) Speaker recognition: a tutorial. Proceedings of the IEEE 85: 1437–1462.

2. Rabiner LR, Juang BH (1993) Fundamentals of speech recognition, volume 14. PTR Prentice Hall Englewood Cliffs.