Laboratory Session 1

Speech is produced by the excitation of time varying vocal tract system by a time varying source (vibrations of vocal cords). The excitation is generated by air flow from lungs carried by trachea through vocal cords. As the acoustic wave passes through the vocal tract, its frequency content (spectrum) is altered by the resonances of the vocal tract. Vocal tract resonances are called formants. Thus, the vocal tract shape can be estimated from the spectral shape (e.g. formant location and spectral tilt) of the speech signal. The speech produced is an acoustic wave which is recorded, sampled, quantized and stored on the computer as sequence of numbers (signal). The speech signal can't be used directly, as the information is in the sequence of the numbers. So the speech signal has to be processed and then features relevant to the task have to be extracted. The features extracted may be related to voice source i.e. vocal cords, like pitch frequency, pitch frequency contour etc. or vocal tract system like linear prediction parameters, cepstral etc. In this laboratory session, we are going to study about speech signal processing and extraction of features related to voice source and vocal-tract system.

The first experiment is to observe a 2 sec speech signal, you will observe that the the energy in speech regions is more than the nonspeech regions. Speech signal is nonstationary in nature so it is processed as short-time signal. In the second experiment, we will select a short-time speech signal and estimate the pitch frequency manually. In the third experiment, we will observe the autocorrelation of the short-time signal and will compute the pitch frequency from it. In the fourth experiment, we will estimate the Fourier spectrum of the short-time signal and will also study the effect of windowing. In the fifth experient, we will study the spectrogram of the speech signal observed in the first experiment. The sixth experiment is about linear prediction analysis and studying the vocal-tract response, like formants and voice source feature like pitch frequency. We will perform linear prediction analysis on different speech sound signals and observe that they have distinct features, in seventh experiment. Though, the features are distinctive in nature they can vary, which makes the tasks such as speech recognition or speaker recognition difficult. We will study about variability introduced by speaker in the eighth experiment. In the last experiment, we will study the pitch contour and the informations embedded in it.

The sampling frequency *sf* of the speech signal is 16000 Hz. Every file name contains information about the gender, speaker, trial number and the sound for e.g. the speech signal file *f_s1_t1_a* means it is the utterance */a/* spoken by female (*f*), (for male its *m* and child *c*) speaker 1 (*s1*) and trial number 1 (*t1*). If you have problems in using any of the routines, use *help* to know the usage, for e.g.

≫ help speech_signal_observation

It will give the usage of the routine *speech_signal_observation*.

Note: The speech files are stored in ascii format so kindly don't edit or tamper it. In all the experiments, the usage of the routine is explained and then an example usage is given. Follow the example usage for now.

1 Speech Signal Observation

Plot the 2 sec speech utterance using the *speech_signal_observation* routine and observe the envelope of the signal. The usage of the routine is as given below

≫ data = speech_signal_observation('filename', fignu, 'plottitle');

this routine plots the speech signal passed to the routine through the file *filename* in a figure numbered *fignu* with the string *plottitle* as the title and returns the speech signal array, which is assigned to the variable *data*. for e.g.

≫ fdata = speech_signal_observation('f_s1_t1_a', 1, 'Utterance /a/ of female speaker 1');

the figure shows the speech utterance plotted in the upper part of the figure and the short-time energy plotted below which is the envelope of the speech signal.

Note: the speech signal array returned will be used for the Experiments 2 and 5.

2 Observation of Short-Time Speech Signal and Manual Pitch Computation

Speech signal is nonstationary in nature but it can be assumed to be quasistationary for one to three pitch periods (short-time signal). In this experiment, we are going to observe a short-time speech signal. Select 30 msec speech signal from the 2 sec speech signal observed in experiment 1 by using routine *select_speech*. The usage of the routine is as given below

≫ stdata = select_speech(data, beginSampleNumber, endSampleNumber, fignu, 'plottitle');

this routine plots a region of the speech signal *data* between the samples *beginSampleNumber* and *endSampleNumber* in figure number *fignu* with the string *plottitle* as the title. For e.g.

≫ fstdata = select_speech(fdata, 15001, 15480, 2, '30 msec Speech Signal of Utterance /a/

Spoken by Female Speaker 1');

Observe the damped sinusoids repeated periodically. Find the period of each sinusoid (neglect the sinusiods which are not complete in the plot) in the following way:

Step 1 Zoom (click) on the largest peak of each sinusoid and note down the sample number (in the x-axis).

Step 2 Find the number of samples between each of the consecutive peaks. It gives the period of each sinusoid.

Average the periods by the number of sinusoids, this is the pitch period, p_t. Calculate the fundamental frequency or pitch frequency F_0 using the following equation (*sf is the sampling frequency*)

$$F_0 = \frac{sf}{p_t} \tag{1}$$

Note: The short-time speech signal region selected in this experiment will be used for the Experiments 3, 4 and 6.

3 Autocorrelation Analysis

In this experiment, we compute the autocorrelation of the short-time speech signal obtained from the Experiment 2 using routine *autocorrelation*. The usage of the routine is as given below

≫ corrdata = autocorrelation(stdata, order, fignu, 'plottitle');

The routine computes the autocorrelation of the short-time signal *stdata* of order *order* and plots it in the figure number *fignu* with string *plottitle* as the title. This routine returns the autocorrelation value to the array *corrdata*. For e.g.

≫ fcorrdata = autocorrelation(fstdata, 256, 3, 'Autocorrelation of order 256 of the 30 msec speech signal of utterance /a/ of female speaker 1');

The length of the autocorrleation array is *order + order + 1* which is symmetric to the point *order + 1* (for the above example it is 257), the value at this point is the energy of the short-time signal for which the autocorrelation was computed. The upper plot shows the actual autocorrelation (observe the symmetricity) and the plot below shows the right half symmetry (i.e. from order + 1 to order + order + 1). Zoom the second peak in this plot and find the sequence number, it is the pitch period p_t. Use equation 1 to find the fundamental frequency F_0. Compare it with the F_0 obtained in the previous experiment.

4 Fourier Spectrum

In this experiment, we compute the Fourier spectrum of the short-time signal *stdata* obtained in the Experiment 2 using routine *FourierSpectrum.* The usage of the routine is as follows

≫ fourierSpectrum(stdata, order, fignu, 'plottitle');

this routine computes DFT of order *order* of the short-time signal *stdata.* The order of the DFT is generally chosen such that it is a 2^n value to use the FFT routine. Depending upon the number of samples select the order of FFT which is near to it, for e.g. 30 msec of the signal we are using has 480 samples so we select order as 512. The Fourier spectrum is plotted in figure *fignu* with string *plottitle* as the title. See the usage e.g. below

≫ fourierSpectrum(fstdata, 512, 4, 'Fourier spectrum of 30 msec speech signal of utterance /a/ of female speaker 1');

the upper plot shows the 512 point DFT spectrum (observe the symmetricity) and the plot below shows the left symmetry of the plot (from point 1 to 256). Observe the spectral peaks they are the formants (resonances in the vocal tract). The 512 point range covers the entire sampling frequency range i.e. 16000 Hz which has redudant information where as the plot below covers half of the sampling frequency i.e. 8000Hz, which is the region of interest (recall the sampling theorem).

5 Spectrogram

In this experiment we are going to compute the narrow-band and wide-band spectrogram of the entire utterance i.e. the signal *data* obtained from the Experiment 1. Recall that in the wide-band we get time resolution and in the narrow-band we get frequency resolution. The spectorgam is computed using *plotSpectrogram* routine. See the usage below

≫ plotSpectrogram(data, order, hamming(order), sf, fignu, 'plottitle');

this routine computes the spectrogram of the given signal *data*, the type of spectrogram depends upon the order *order*, for wide-band spectrogram we need small window we choose order 256 or 128 which is a short duration, whereas, for narrow-band we choose order as 1024 or 2048, which is a long duration so we loose the time resolution. *hamming* is the window and *sf* sampling frequency. The spectrogram is plotted in the figure *fignu* with the string plottitle as the title. See the usage examples below

Wide Band Spectrogram
≫ plotSpectrogram(fdata, 256, hamming(256), 16000, 6, 'wide band spectrogram of the utterance /a/ of female speaker 1');

Narrow Band Spectrogram

$\gg$ plotSpectrogram(fdata, 1024, hamming(1024), 16000, 7, 'narrow band spectrogram of the utterance /a/ of female speaker 1');

The upper plot shows the time domain speech signal and the lower plot shows the spectrogram of the time domain speech signal in the upper plot. The high energy regions are in red color, the more dark it is the more energy is in that region.

6 Linear Prediction (LP) Analysis

Linear prediction is the most common technique to estimate the shape of the vocal tract. A *pth* order linear prediction expresses every sample as the linear weighted sum of the past p samples. The resulting difference equation expressed in $z - domain$ is

$$H(z) = \frac{1}{1 - \sum_{j=1}^{p} a_j z^{-j}} \tag{2}$$

The idea behind linear prediction analysis is to estimate the p a_ks' which minimize the prediction error in mean-square sense. The linear prediction error is also called LP residual. The a_ks' determine the solution of the equation. The solution of the equation in denominator is called pole. A real pole determines the spectral roll off and a complex pole (which always exist with a conjugate) determines the location of the formant in the LP spectrum. The LP spectrum is the Fourier transform of the a_ks'.

6.1 LP Spectrum

In this experiment, we will observe the LP spectrum of the short-time speech signal obtained from Experiment 2 using routine *lpSpectrum*. The usage of the routine is as given below

$\gg$ lpSpectrum(stdata, lporder, win, order, sf, fignu, 'plottitle')

stdata is the short-time signal obtained from *select_speech* routine, *lporder* is the linear prediction order p, to window the signal by a hamming window set *win* 1 else set it 0 (rectangular window), *order* is the FFT order needed to compute the linear prediction spectrum from a_ks', *sf* is the sampling frequency (which is 16000). The resulting linear prediction and Fourier spectrum are plotted in figure number *fignu* with *plottitle* as the title of the plot. e.g.

$\gg$ lpSpectrum(fstdata, 14, 1, 512, 16000, 8, 'Linear Prediction Spectrum of the short-time signal fstdata')

In the figure, you will observe two plots. The upper plot is the Fourier spectrum and the lower plot is the linear prediction spectrum. Observe the spectral peaks (formants) in the linear prediction spectrum (which is very clear) and Fourier spectrum. Zoom in the spectral peaks and note down the frequency displayed on the x-axis then zoom in the wideband spectrogram (observed in the earlier experiment) near that spectral frequency and observe that the energy is indeed high in that region. Now change the linear prediction order i.e. *lporder* to, say 1, 3, 16, 20, 30, 50 and observe the changes in the LP spectrum. Try to reason it.

6.2 LP Residual

In this experiment we will perform linear prediction analysis and compute the LP residual of short-time speech signal obtained from Experiment 2. The usage of the routine is as given below.

≫ residual = lpResidual(stdata, nsample, lporder, fignu, 'plottitle')

stdata is the short-time signal obtained from *select_speech* routine, *nsample* is the number of samples in the short-time signal, *lporder* is the linear prediction order *p*. The routine plots the short-time signal and the LP Residual of it in the figure *fignu* with title *plottitle*. An example of the usage is as given below

≫ residual = lpResidual(fstdata, 480, 10, 9, 'LP Residual Signal');

Step 1 Zoom (click) on the largest peak of each sinusoid in the upper plot and note down the sample number (in the x-axis).

Step 2 Zoom (click) on the corresponding peaks in the LP residual signal in the upper plot and note down the sample number (in the x-axis).

Step 3 Compare the observations of step 1 and 2. Are they same?

Perform an autocorrelation analysis on the residual signal using routine *autocorrelation* and find the pitch period as it was done in the Experiment 3. Usage of the routine is as given below.

≫ autocorrelation(residual, 256, 10, 'Autocorrelation of LP Residual signal');

7 LP Spectrum of Different Speech Sounds

In the Experiment 6.1, we studied the LP spectrum of short-time signal. In this experiment, we are going to study the LP spectrum of different speech sounds using routine *lpSpectrum_Sounds*. The usage of the routine is as given below

≫ lpSpectrum_Sounds(fignu)

This routine computes the LP spectrum of sounds */a/, /e/, /i/, /o/, /u/* and plots them in different figures. The input to this routine is figure number *fignu*. An example usage is given below

≫ lpSpectrum_Sounds(1)

Note down your observation.

8 Intra and Inter Speaker Variability

In the Experiments 6 and 7, we studied the effect of order on linear prediction and also we observed that for different sounds the formants are different. In this experiment, we are going to study about variability caused by speakers. There are two kinds of speaker variability that are of interest, they are intra-speaker variability and inter-speaker variability. Intra-speaker variability is the variability introduced by the same speaker while producing the same sound. Inter-speaker variability is the variability introduced by different speakers when producing the same sound. This can be useful depending upon type of application, such as in speech recognition it is good if there is no speaker variability, where as, for speaker recognition inter-speaker variability is very important. Intra-speaker variability is neither useful for speech recognition nor for speaker recognition applications.

1. The first experiment is to study the intra-speaker variability. Three utterances of the same sound */a/* spoken by the same speaker at three different instants is used for this study. For this study, we will use the routine *speakerVariation*. This routine takes in the utterances file names, their corresponding begin and end points defining the short-time signal and figure number as input. See the usage below.

   ≫ speakerVariation('f_s2_t1_a', 'f_s2_t2_a', 'f_s2_t3_a', 14001, 14480, 10001, 10480, 12481, 12960, 1);

   The linear prediction spectrum of the short-time signal of all the three utterances is computed and plotted in the same figure. Observe that the first two formants regions for all the three utterances are almost same but it is not the case with higher formants.

2. The second experiment is to study the inter-speaker variability. Three utterances of sound */a/* spoken by a female, male and child is used for this study. The same routine *speakerVariation* is used for this study as shown below.

   ≫ speakerVariation('f_s1_t1_a', 'm_s2_t1_a', 'c_s1_t1_a', 15001, 15480, 9001, 9480, 12481, 12960, 1);

The linear prediction of the short-time signal of all the three utterances is plotted in the same figure. Again observe that the first two formant regions for the male and female speaker are almost same, in case of child speech the second formant is very much shifted than the first formant. Like in the previous experiment we observe that the higher formant regions are different for different speakers even though the same sound $/a/$ is being spoken.

9 SIFT Algorithm and Pitch Contour

In this experiment, we extend the idea of pitch estimation using LP residual (refer to Experiment 6.2) into a pitch estimation algorithm. The pitch frequency can be estimated through Simple Inverse Filter Tracking (SIFT) algorithm. The SIFT algorithm computes the pitch frequency for a given short-time speech signal in the following way.

Step 1 Low pass filter the short-time signal.

Step 2 Perform linear prediction analysis and obtain the linear prediction residual.

Step 3 Perform autocorrelation on the linear prediction residual.

Step 4 Find the location of second peak, make a decision on voicing. If it is voiced compute the pitch frequency else set pitch frequency to zero.

The pitch frequency contour for a spoken sentence can be computed by taking a short-time window of size say 30 msec.

1. Place this window at the begining of the speech signal compute the pitch frequency using the SIFT algorithm.
2. Shift the window by 10 msec on the speech signal and compute the pitch frequency using SIFT algorithm.
3. Repeat Step 2 until the end of the speech signal is reached.

The 10 msec shift is called as frame. So we obtain a pitch frequency for every 10 msec or frame. The pitch contour is nothing but the array of pitch frequencies obtained for the sequence of frames. For applications like speech or speaker recognition for every frame a feature parameter vector (e.g. linear prediction coefficients a_ks') is obtained. In other words, the feature extraction stage yields a sequence of feature parameter vectors $x_1, x_2 \cdots x_{N-1}, x_N$, where N is the number of frames.

In this experiment, first, we are going to observe the pitch contour for two different types of sentences, interrogative and declarative, using routine *sift*. For this study, we will use 30 msec frame size (480 samples), a shift of 10 msec (160 samples) and linear prediction order of 10. The usage is as given below.

The sentence spoken is an interrogative sentence, *Where are you from?*

```
≫ sift(’m_s1_i_sen1’, 480, 160, 10, 16000, 1, ’Pitch contour of interogative sentence spoken
by male speaker 1’);
```

In the figure 1, you can observe the window moving across the signal in the upper plot and in the lower plot the corresponding LP residual signal. Once the estimation of the pitch contour is over, the speech signal (upper plot) and the pitch contour (lower plot) are plotted in figure 2. Observe the rise and fall of pitch contour across the sentence. This rise and fall of pitch contour carries information like speaking style, type of sentence, emotional status of speaker etc. Observe the rise of pitch contour for the word *where* at the begining of the sentence (in the context of interogation). If a line is drawn interpolating the peaks and valleys in the pitch contour, it will have a positive slope. Observe at the end again a fall and then rise of pitch contour.

The sentence spoken is a declarative sentence, *I am from India.*

```
≫ sift(’m_s1_d_sen1’, 480, 160, 10, 16000, 3, ’Pitch contour of declarative sentence spoken
by male speaker 1’);
```

Again in the figure 3 you can observe the window moving across the signal in the upper plot and in the lower plot the corresponding LP residual signal. Once the estimation of the pitch contour is over, the speech signal (upper plot) and the pitch contour (lower plot) are plotted in figure 8. Observe the rise and fall of pitch contour across the sentence. If a line is drawn interpolating the peaks and valleys in the pitch contour, it will have a negative slope.

Note that the pitch frequency for a single frame is just an information about the speaker. It doesnot convey any information regarding the sentence being spoken or message or emotional status of speaker. But when a large duration (say 100-300 msec) i.e. a sequence of frames are considered then we can observe the rise and fall of pitch contour and get the informations related to it. Still pitch contour does not conveys any information regarding the message being spoken. Now we will perform the pitch contour analysis on the same sentences spoken by a different speaker

```
≫ sift(’m_s2_i_sen1’, 480, 160, 10, 16000, 5, ’Pitch contour of interogative sentence spoken
by male speaker 2’);
```

```
≫ sift(’m_s2_d_sen1’, 480, 160, 10, 16000, 7, ’Pitch contour of declarative sentence spoken
by male speaker 2’);
```

Compare the pitch contour in plots 2 and 6. They are the pitch contour of the same interogative sentence spoken by two different speakers. Are they different? Similary compare the plots 4 and 8 and put down your observations. Human use efficiently the speaking style information which is embedded in the pitch contour to recognize another person.