Speech feature extraction

Speech feature extraction is the mathematical representation of the speech file, which converts the speech waveform to some type of parametric representation for further analysis and processing in speech recognition. A good feature may produce a good result for any recognition system. It transforms the processed speech signal to a concise but logical representation that is more discriminative and reliable than the actual signal.

The most important parametric representation of speech is the short time spectral envelope and the spectral analysis method is the core of the signal processing front-end in a speech recognition system.

Two important types of speech features, such as time-domain signal features and frequency-domain signal features, have been used in speech processing.

Speech feature extraction techniques:

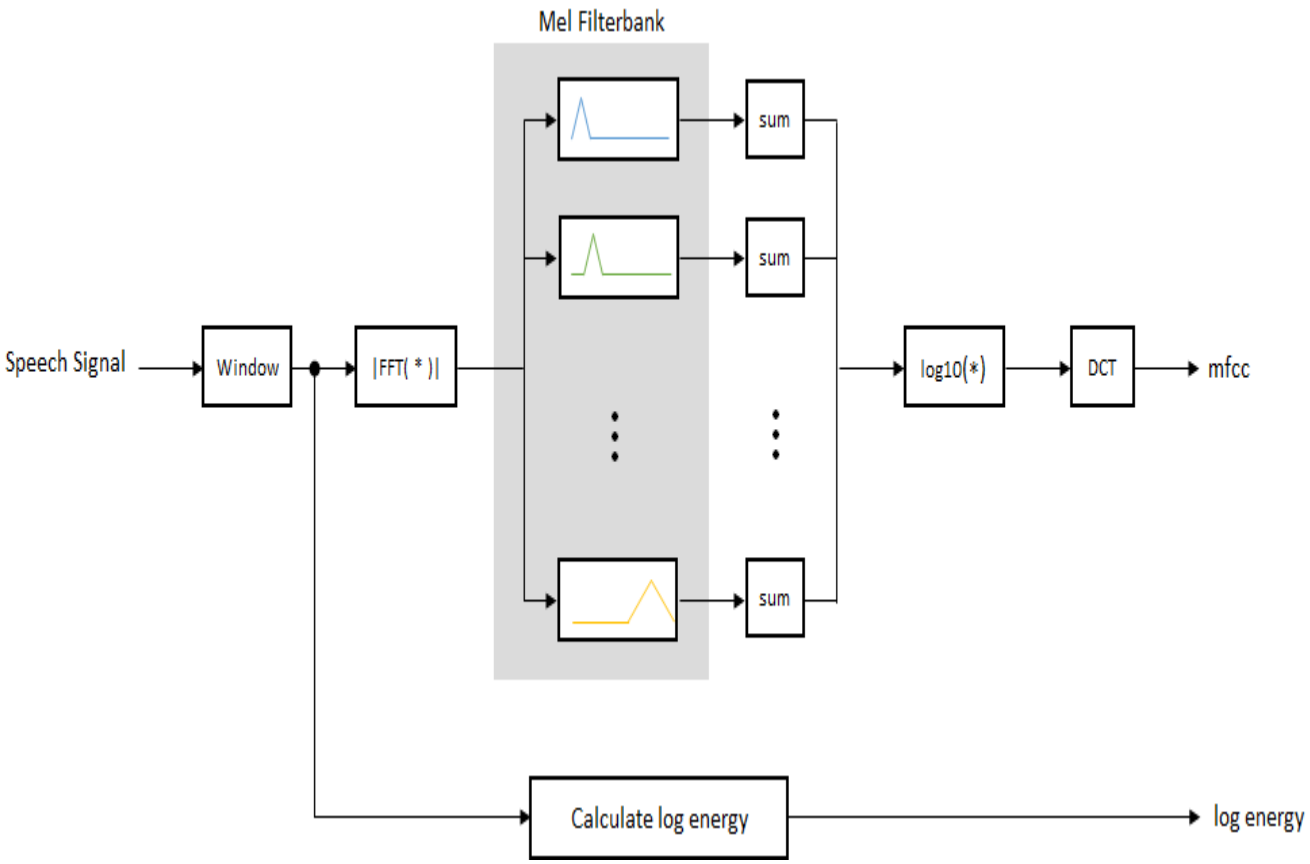
1. Short time energy
2. Zero crossing rates
3. level crossing rates
4. spectral centroid
5. Mel Frequency Cepstral Coefficients (MFCC)
6. Linear Prediction Coefficients (LPC)
7. Linear Prediction Cepstral Coefficients (LPCC)
8. Line Spectral Frequencies (LSF)
9. Discrete Wavelet Transform (DWT)
10. Perceptual Linear Prediction (PLP)

Mel-Frequency Cepstrum Coefficients (MFCC)

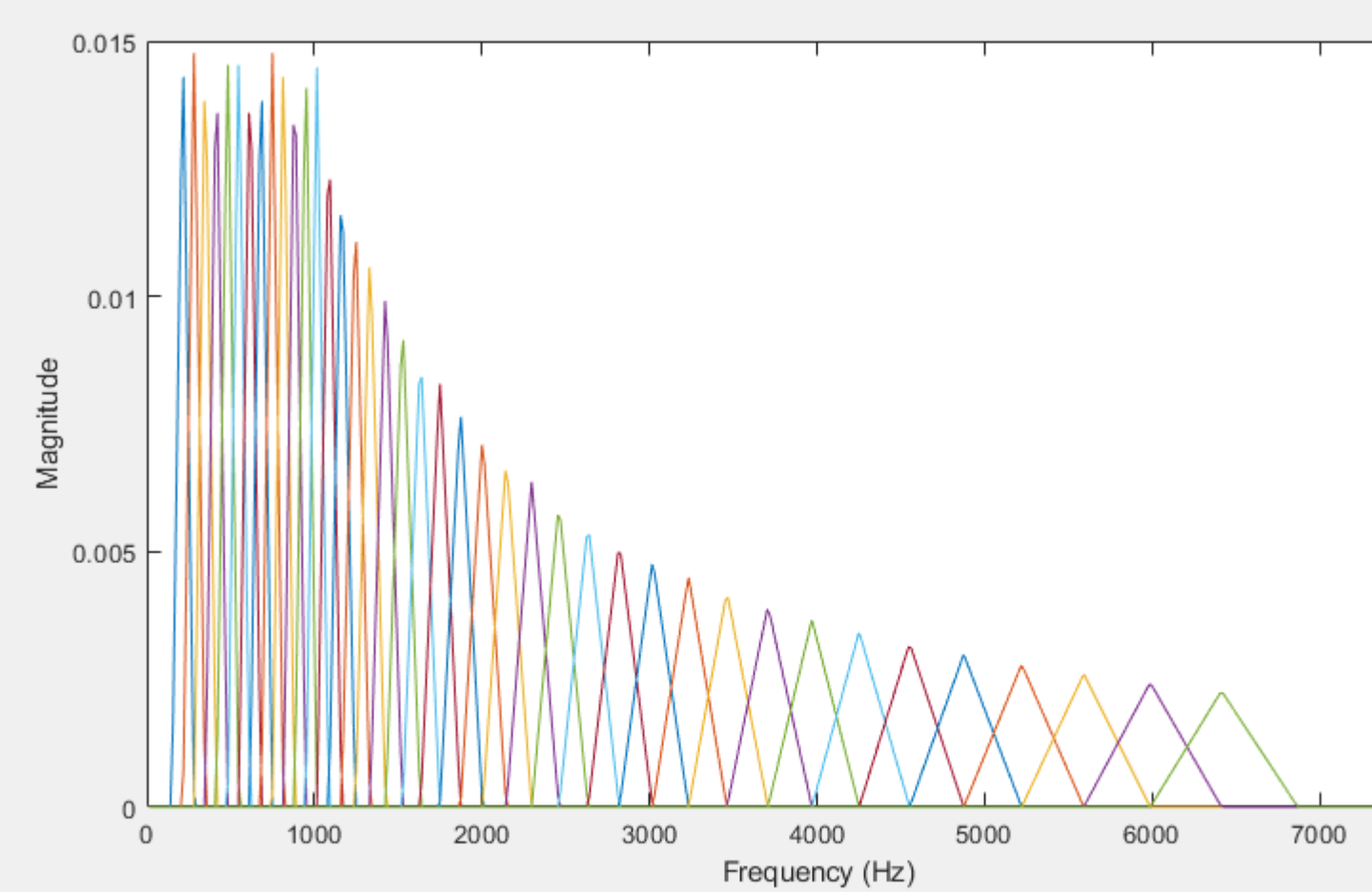
MFCC are popular features extracted from speech signals for use in recognition tasks. In the source-filter model of speech, MFCC are understood to represent the filter (vocal tract). The frequency response of the vocal tract is relatively smooth, whereas the source of voiced speech can be modeled as an impulse train. The result is that the vocal tract can be estimated by the spectral envelope of a speech segment.

The motivating idea of MFCC is to compress information about the vocal tract (smoothed spectrum) into a small number of coefficients based on an understanding of the cochlea.

Although there is no hard standard for calculating MFCC, the basic steps are outlined by the diagram.



The mel filterbank linearly spaces the first 10 triangular filters and logarithmically spaces the remaining filters. The individual bands are weighted for even energy. The graph represents a typical mel filterbank.



MFCC computation is a replication of the human hearing system intending to artificially implement the ear's working principle with the assumption that the human ear is a reliable speaker recognizer. MFCC features are rooted in the recognized discrepancy of the human ear's critical bandwidths with frequency filters spaced linearly at low frequencies and logarithmically at high frequencies have been used to retain the phonetically vital properties of the speech signal.

Speech signals commonly contain tones of varying frequencies, each tone with an actual frequency, f (Hz) and the subjective pitch is computed on the Mel scale. The mel-frequency scale has linear frequency spacing below 1000 Hz and logarithmic spacing above 1000 Hz. Pitch of 1 kHz tone and 40 dB above the perceptual audible threshold is defined as 1000 mels, and used as reference point.

MFCC is used to identify airline reservation, numbers spoken into a telephone and voice recognition system for security purpose.

Some modifications have been proposed to the basic MFCC algorithm for better robustness, such as by lifting the log-mel-amplitudes to an appropriate power (around 2 or 3) before applying the DCT and reducing the impact of the low-energy parts.

Algorithm

In the computation of MFCC,

The first thing is windowing the speech signal to split the speech signal into frames. Since the high frequency formants process reduced amplitude compared to the low frequency formants, high frequencies are emphasized to obtain similar amplitude for all the formants.

After windowing, Fast Fourier Transform (FFT) is applied to find the power spectrum of each frame.

Subsequently, the filter bank processing is carried out on the power spectrum, using mel-scale.

The DCT is applied to the speech signal after translating the power spectrum to log domain in order to calculate MFCC coefficients.

The formula used to calculate the mels for any frequency is:

$$mel(f) = 2595xlog10(1 + f/700)$$

where mel(f) is the frequency (mels) and f is the frequency (Hz).

The MFCCs are calculated using this equation:

$$\hat{C}_n = \sum_{n=1}^{k} (log\, \hat{S}_k) \cos\left[n\left(k - \frac{1}{2}\right)\frac{\pi}{k}\right]$$

where k is the number of mel cepstrum coefficients, $\hat{S}_k$is the output of filterbank and $\hat{C}_n$ is the final mfcc coefficients.

The block diagram of the MFCC processor can be seen in Figure 1. It summarizes all the processes and steps taken to obtain the needed coefficients. MFCC can effectively denote the low frequency region better than the high frequency region, henceforth, it can compute formants that are in the low frequency range and describe the vocal tract resonances. It has been generally recognized as a front-end procedure for typical Speaker Identification applications, as it has reduced vulnerability to noise disturbance, with minute session inconsistency and easy to mine [19]. Also, it is a perfect representation for sounds when the source characteristics are stable and consistent (music and speech). Furthermore, it can capture information from sampled signals with frequencies at a maximum of 5 kHz, which encapsulates most energy of sounds that are generated by humans.

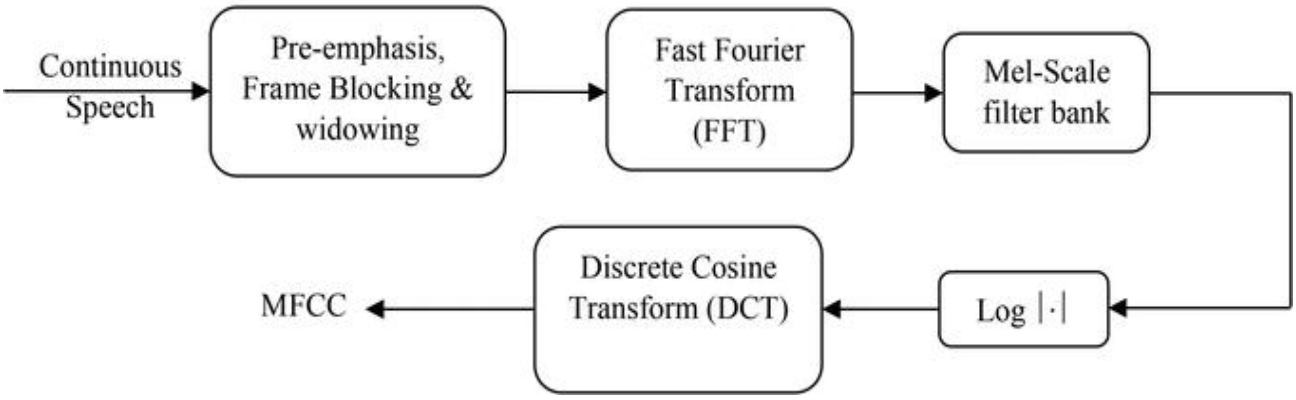
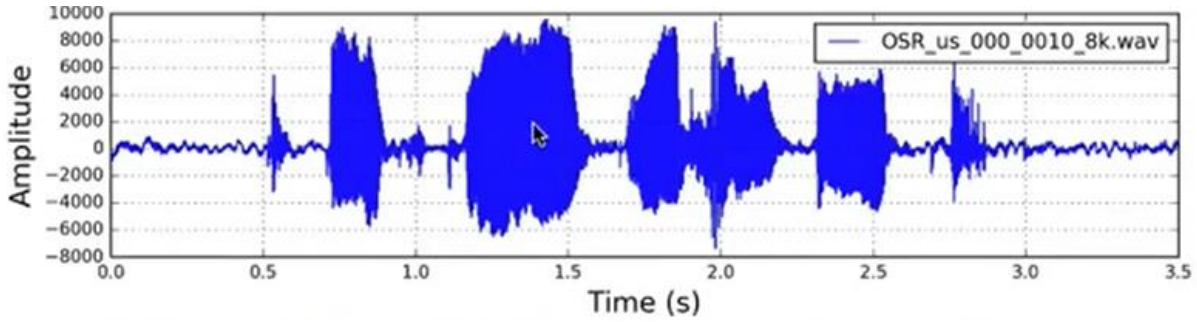


Figure 1. Block diagram of MFCC processor.

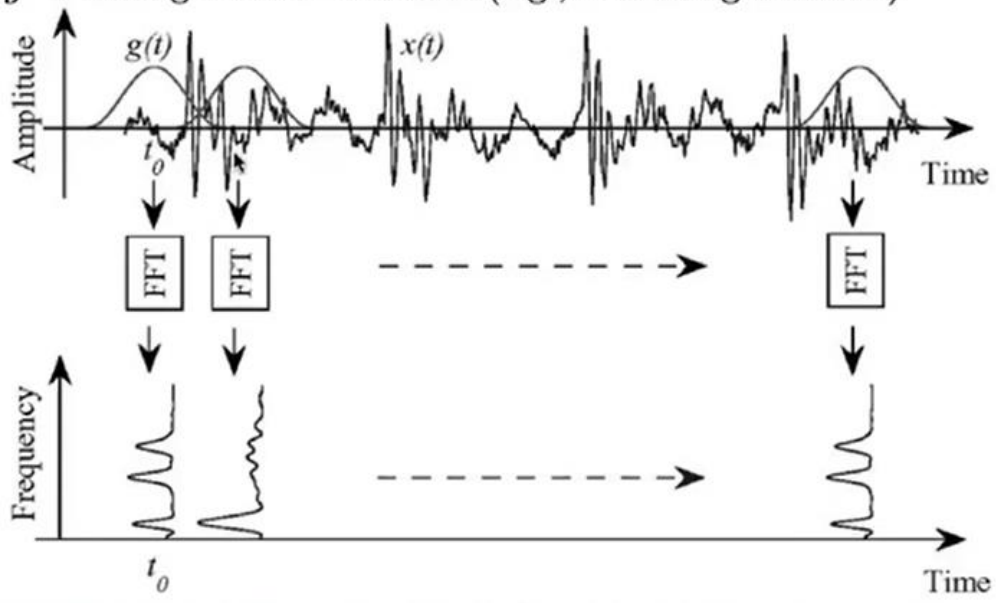


Short-Time Fourier-Transform (STFT)

$x \rightarrow$ signal in the time domain (e.g., $x \in \mathbb{R}^{28{,}200=3.525 \text{ seconds} \times 8{,}000 \text{ Hz}}$)

sampling frequency (number of samples per seconds)

$g \rightarrow$ sliding window function (e.g., Hamming function)



$X_i \in \mathbb{R}^{200} \rightarrow i$-th frame of signal x (25ms frames)

$80 \rightarrow$ frame step (10ms) $\implies X \in \mathbb{R}^{200\times 350}$ $(350 = (28,200 - 200)/80)$

$\widetilde{X}_i \in \mathbb{C}^K \rightarrow$ discrete Fourier transform of $X_i \implies \widetilde{X} \in \mathbb{C}^{K\times 350}$

$$\widetilde{X}_i(k) = \sum_{n=1}^{N} X_i(n) g(n) e^{-j2\pi kn/N}, k = 1, \ldots, K \qquad N = 200$$

$K = 257 \rightarrow$ number of discrete Fourier transform coefficients

$P_i(k) = \frac{1}{N}|\widetilde{X}_i(k)|^2 \rightarrow$ Periodogram estimate of the power spectrum

$\implies P \in \mathbb{R}^{257\times 350}$

Mel-spaced Filterbank

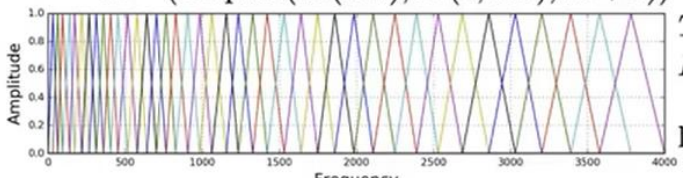
300 Hz $\rightarrow$ lower frequency

4,000 Hz $\rightarrow$ upper frequency

$M(f) = 1,125 \ln(1 + f/700) \rightarrow$ convert frequency to Mel scale

$M^{-1}(m) = 700(\exp(m/1,125) - 1) \rightarrow$ convert Mel scale to Hz

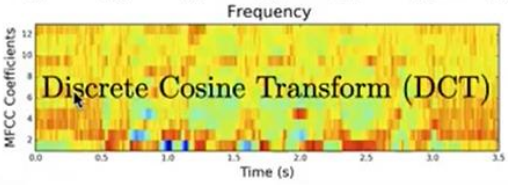
$M^{-1}(\text{linspace}(M(300), M(4,000), 26 + 2))$ $26 \rightarrow$ number of triangular filters



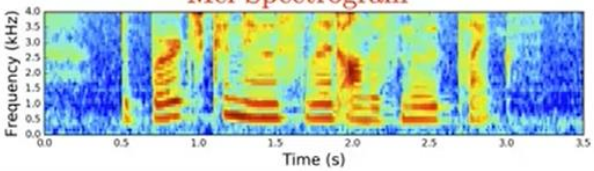
$T \in \mathbb{R}^{26\times 257} \implies E = TP \in \mathbb{R}^{26\times 350}$

$E_i(l) \rightarrow$ amount of energy in filter bank l at frame i

$\log(E) \in \mathbb{R}^{26\times 350} \rightarrow$ log filter bank energy



Mel Spectrogram



Short Term Energy Parameter

The energy associated with speech is time varying in nature. Hence the interest for any automatic processing of speech is to know how the energy is varying with time and to be more specific, energy associated with short term region of speech. By the nature of production, the speech signal consist of voiced, unvoiced and silence regions. Further the energy associated with voiced region is large compared to unvoiced region and silence region will not have least or negligible energy. Thus short term energy can be used for voiced, unvoiced and silence classification of speech.

The relation for finding the short term energy can be derived from the total energy relation defined in signal processing. The total energy of an energy signal is given by

$$E_T = \sum_{n=-\infty}^{\infty} s^2(n) \qquad (1)$$

In case of short term energy computation we consider speech in terms of 10-30 msec. Let the samples in a frame of speech are given by ***"n=0 to n=N-1"***, where ***" N "*** is the length of frame (samples), then for energy computation the speech will be zero outside the frame length. Then for energy computation amplitude of the speech samples will be zero outside the frame. Accordingly we can write above mentioned relation as

$$E_T = \sum_{n=-\infty}^{-1} s^2(n) + \sum_{n=0}^{N-1} s^2(n) + \sum_{n=N}^{\infty} s^2(n)$$

$$E_T = \sum_{n=0}^{N-1} s^2(n) \qquad (2)$$

This relation will give total energy present in the frame of speech from ***" n=0 to n=N-1 "***. To represent more specifically, only one frame of speech we use the relation

$$s_w(n) = s(m).w(n-m)$$

where "***w(n)***" represent the windowing function of finite duration. There are several windowing functions present in the signal processing literature. The mostly used ones include rectangular, hanning and hamming. For all time domain parameters estimation we use the rectangular window for its simplicity.

Now we can write the relation of short term energy as follows

where ***"n"*** is the shift / rate in number of samples at which we are interested in knowing the short term energy. The shift can be as small as one sample or as large as frame size. The short term energy computed for every sample shift may not be required since the energy variation in case of speech is relatively slow. For this reason the shift is kept much larger than one sample. Usually it is about half the frame size.

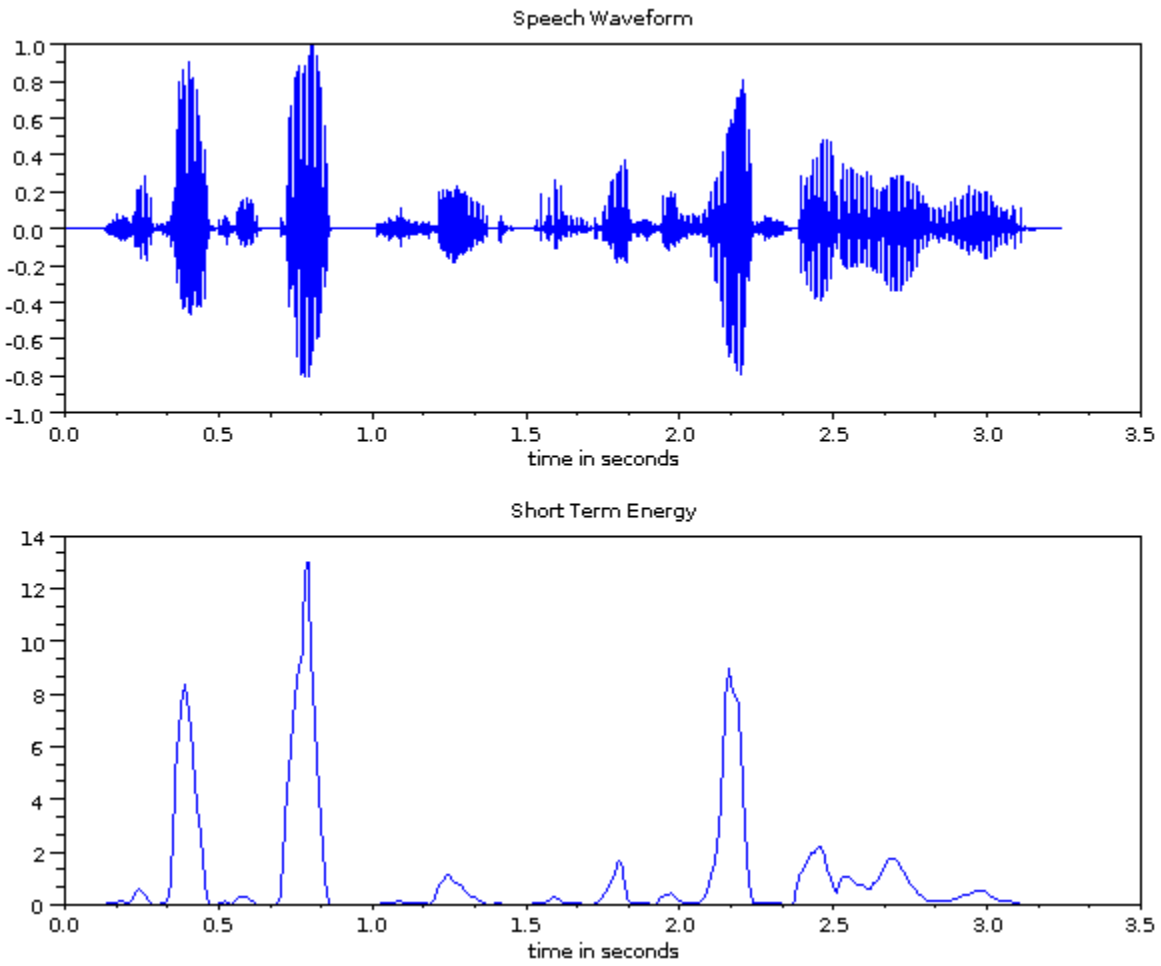


Figure_1: Short term energy contour for the spech signal

The last point about the short term energy is the value for frame size. Since the stationary assumption in case of speech is valid for 10 to 30 msec, the typical value for the frame size is about 20 msec. Alternatively, for larger frame sizes we get much smoothed version of energy and may not find time varying nature of short term energy. Figure_1 shows the energy contours for speech signal taken for study .

Short time magnitude is similar to short time energy where the weighted sum of absolute values of the signal is computed instead of sum of the squares

Short Term Zero Crossing Rate (ZCR)

Zero Crossing Rate gives information about the number of zero-crossings present in a given signal. Intuitively, if the number of zero crossings are more in a given signal, then the signal is changing rapidly and accordingly the signal may contain high frequency information. On the similar lines, if the number of zero crossing are less, hence the signal is changing slowly and accordingly the signal may contain low frequency information. Thus ZCR gives an indirect information about the frequency content of the signal.

The ZCR in case of stationary signal is defined as,

$$z = \sum_{n=-\infty}^{\infty} |sgn(s(n)) - sgn(s(n-1))|$$

$$where\ sgn(s(n)) = 1\ if\ s(n) \geq 0$$

$$= -1\ if\ s(n) < 0$$

This relation can be modified for non-stationary signals like speech and termed as short term ZCR. It is defined as

$$z(n) = \frac{1}{2N} \sum_{m=0}^{N-1} s(m).w(n-m)$$

The factor "2" comes in the denominator to take care of the fact that there will be two zero crossings per cycle of one signal

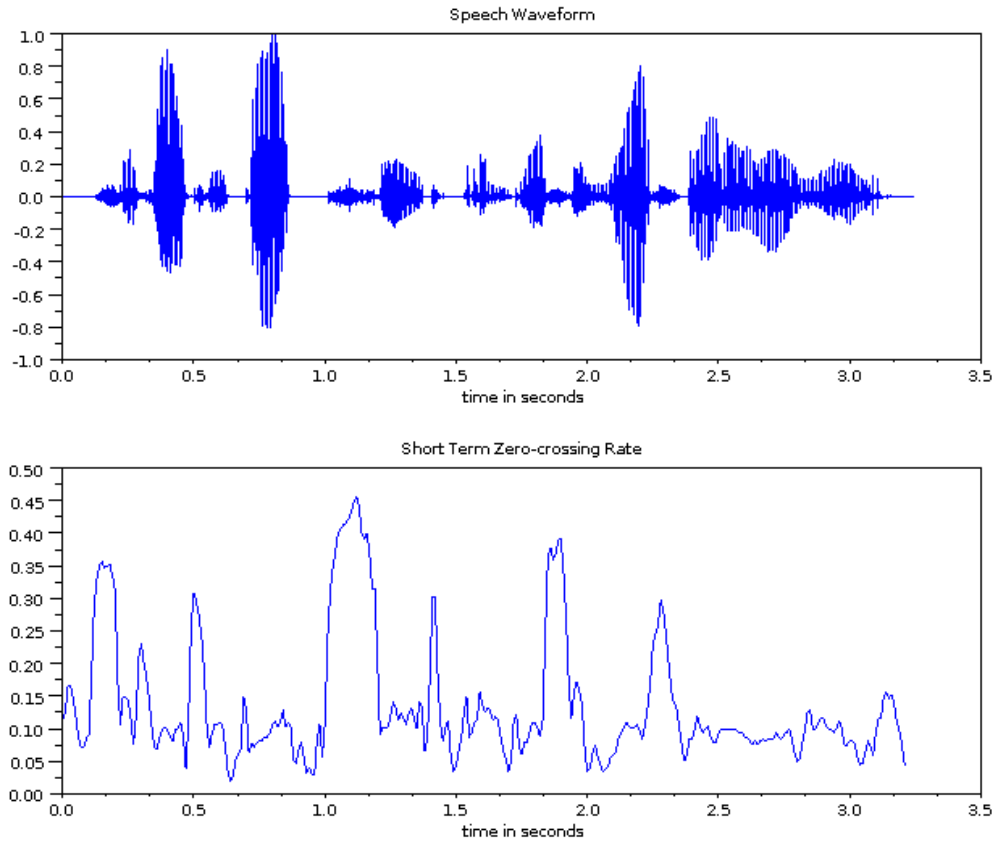


Figure_2: Short term zero crossing rate of a speech signal

In case of speech the nature of signal changes with time over few msec. For instance, from initial voiced to unvoiced and back to voiced and so on. To have some useful information, ZCR needs to be computed using typical frame size of 10-30 msec with half the frame size as shift. A speech signal for the message **" she had your suit in your greasy wash water all year**" and its short term ZCR computed are shown in Figure_2. As it can be observed, in case of unvoiced sounds like |s|, the ZCR value is significantly high compared to the region of voiced sounds like |a| and hence can be used for distinguishing voiced and unvoiced regions.

Uses of short time zero crossing rate: Zero Crossing Rate gives information about the number of zero-crossings present in a given signal. Intuitively, if the number of zero crossings is more in a given signal, then the signal is changing rapidly and accordingly the signal may contain high frequency information. On the similar lines, if the number of zero crossing is less, hence the signal is changing slowly and accordingly the signal may contain low frequency information. Thus ZCR gives indirect information about the frequency content of the signal.

Human beings express their feelings, opinions, views and notions orally through speech. The speech production process includes articulation, voice, and fluency. It is a complex naturally acquired human motor abilities, a task categorized in regular adults by the production of about 14 different sounds per second via the harmonized actions of roughly 100 muscles connected by spinal and cranial nerves.

Short time analysis:

Why do we consider short time representation of speech signals?

Because speech sounds change over time, we need to analyze only short regions of the signal. We convert the speech signal into a sequence of frames.
Short-term analysis is the first step that takes us out of the time domain and into some other domain, such as the frequency domain.

Apart from measuring the total duration, it makes no sense to analyze any other properties of a whole utterance.

Speech signals are nonstationary in nature, means their statistical parameters like intensity and variance vary over time. Speech signals may be stationary for a shorter period but when considered over a longer duration they are aperiodic. Therefore the Fourier transform is not a suitable technique for speech analysis as it requires a periodic signal for infinite time. A technique called short time analysis is used. In this technique, the signal is divided into short frames or segments, assuming the signal is stationary in that short frame and analyzing each frame or segment separately. The length of each frame is about 10–20 ms, short enough to satisfy the assumption.

The spectrogram is an example of short rime analysis which is discussed in the next section.

Discrete-Time Fourier Transform is applied to each frame which results in spectra over time.
For a given signal x[n], the short time signal xm[n] of frame m is represented by Eq. (10.15):

$$x_m[n] = x[n].w_m[n]$$

where wm[n] is a window function, beyond a specific region its value is zero. For this technique, wm[n] should be equal for all frames. So the calculations are as in Eqs. (10.16) and (10.17).

$$w_m[n] = w[m-n]$$

$$w[n] = \begin{Bmatrix} \widehat{w}[n] & |n| \le \frac{N}{2} \\ 0 & |n| > \frac{N}{2} \end{Bmatrix}$$

where N is the length of the window.

There are two useful acoustic features in a voiced-speech signal: fundamental frequency (pitch) and formant. The fundamental frequency is usually the lowest frequency component of the signal; it represents the vibration frequency of the vocal cords during sound production. The formant is a concentration of acoustic energy around a particular frequency in the speech wave; each formant corresponds to a resonance on the vocal tract.

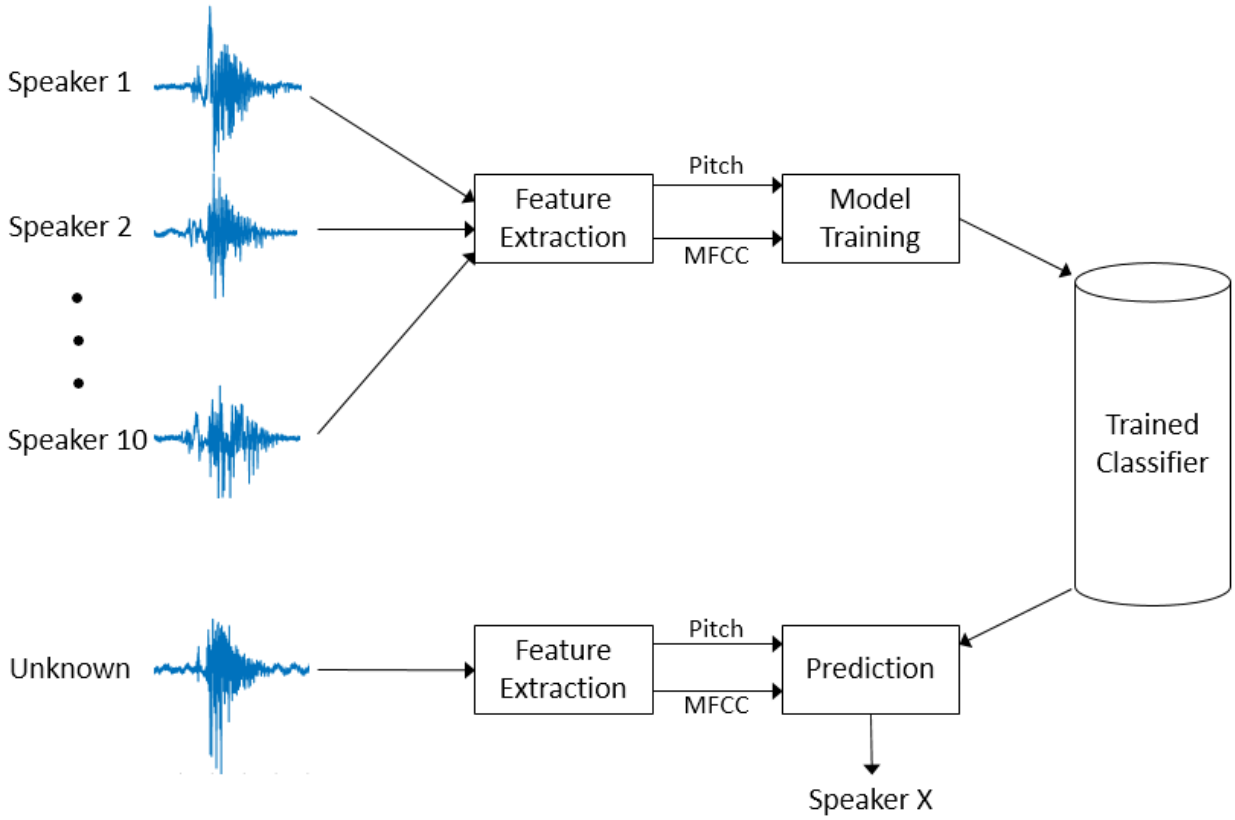
Speaker Identification Using Pitch and MFCC - MATLAB & Simulink (mathworks.com)

Speaker Identification Using Pitch and MFCC

This example demonstrates a machine learning approach to identify people based on features extracted from recorded speech. The features used to train the classifier are the pitch of the voiced segments of the speech and the mel frequency cepstrum coefficients (MFCC). This is a closed-set speaker identification: the audio of the speaker under test is compared against all the available speaker models (a finite set) and the closest match is returned.

Introduction

The approach used in this example for speaker identification is shown in the diagram.



Pitch and MFCC are extracted from speech signals recorded for 10 speakers. These features are used to train a K-nearest neighbor (KNN) classifier. Then, new speech signals that need to be classified go through the same feature extraction. The trained KNN classifier predicts which one of the 10 speakers is the closest match.

Features Used for Classification

This section discusses pitch, zero-crossing rate, short-time energy, and MFCC.

- ✓ Pitch and MFCC are the two features that are used to classify speakers.
- ✓ Zero-crossing rate and short-time energy are used to determine when the pitch feature is used.

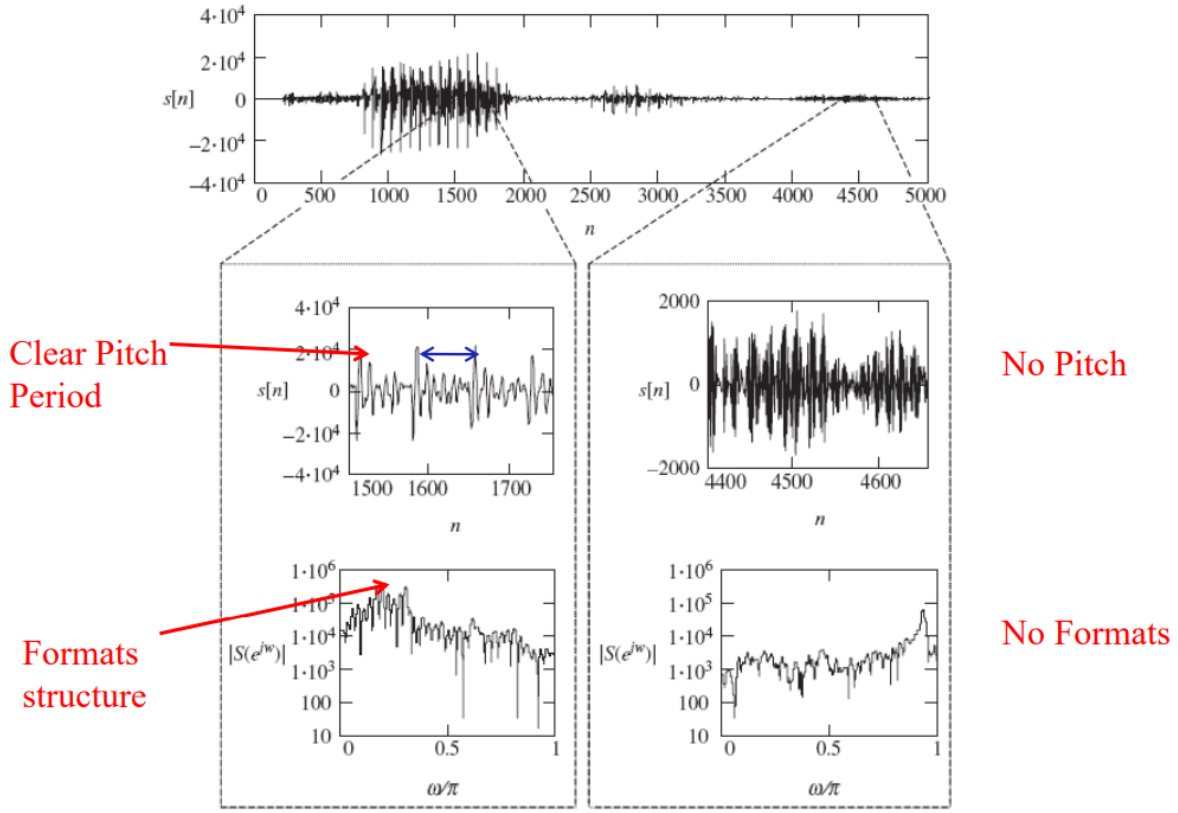
Pitch

Speech can be broadly categorized as *voiced* and *unvoiced*.

- ✓ In the case of voiced speech, air from the lungs is modulated by vocal cords and results in a quasi-periodic excitation. The resulting sound is dominated by a relatively low-frequency oscillation, referred to as *pitch*.
- ✓ In the case of unvoiced speech, air from the lungs passes through a constriction in the vocal tract and becomes a turbulent, noise-like excitation. In the source-filter model of speech, the excitation is referred to as the source, and the vocal tract is referred to as the filter. Characterizing the source is an important part of characterizing the speech system.

As an example of voiced and unvoiced speech, consider a time-domain representation of the word "two" (/T UW/). The consonant /T/ (unvoiced speech) looks like noise, while the vowel /UW/ (voiced speech) is characterized by a strong fundamental frequency.

The pitch is a subjective attribute of the speech and is related to the fundamental frequency of the voiced speech signal. Pitch estimation is important in many areas of speech processing. Pitch estimation is necessary for coding and recognition of speech. For example, it is used in modern speech coders technology for the hearing impaired and speaker recognition systems.



Example of speech waveform (male) of the word "problems."

Estimation pitch parameters using cepstrum

1) Time Domain: Pitch refers to the fundamental frequency of a voice signal. It evaluates how much stress is applied while generating the voice. To detect pitch in the time domain, autocorrelation is the most popular and effective technique developed so far. Autocorrelation is

the cross-correlation of a speech signal with itself in the time domain. At first, the down sampled speech is split into a 40 ms window segment and the corresponding samples are read using the 'audioread' function in MATLAB. Then we invoke the built-in function 'xcorr' which returns the cross-correlation sequence of the window segment of the sampled signal. The pitch period is then found by calculating the time lag between the central peak and the second highest peak of the autocorrelation sequence. Finally, pitch frequency is calculated simply by taking inverse of the pitch period.

2) Frequency Domain: To detect pitch in frequency domain, the cepstrum method is a widely used algorithm. Cepstrum of a signal is obtained by taking the Inverse Fourier Transform (IFT) of the logarithm of the spectrum of that signal. Mathematically where x[n] is the sampled speech signal, F indicates its Fourier transform, and c[n] are the cepstrum coefficients. Like the autocorrelation method, the cepstrum method also reads 40 ms window segment of the down sampled Bangla voice signal using the 'audioread' function in MATLAB. Then the signal is multiplied by a hamming window. Fast Fourier Transformation (FFT) of this windowed frame gives the spectrum of the speech signal in the frequency domain. Taking Inverse Fourier Transformation (IFT) of the logarithm of the spectrum gives cepstrum in the quefrency domain. Once in the quefrency domain, the pitch can be estimated by determining the peak of the cepstrum which represents pitch lag. The lag at which there is the most energy represents the dominant frequency in the log spectrum and thereby it gives the pitch frequency. A flow diagram of cepstrum algorithm is shown in below:

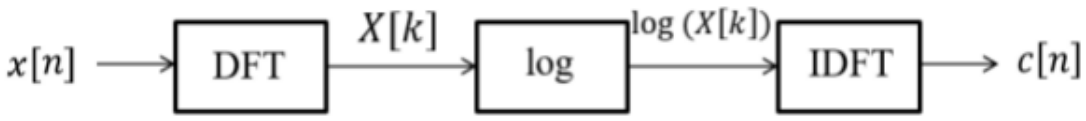


Figure- 2: Flow diagram of cepstrum algorithm for pitch Estimation.

Pitch estimation using cesptrum analysis:

(i) The speech signal gives as input to system consists of periodic excitation convolved with the impulse response of the vocal tract which is slowly varying function.

(ii) The FFT block takes the DFT of a signal to obtained the spectrum of the signal. When we take the log magnitude, we get amplitude calculation in dB.

(iii) It can be seen that the periodic excitation is rapidly varying and the vocal tract response, which is the envelop of the plot, is slowly varying function.

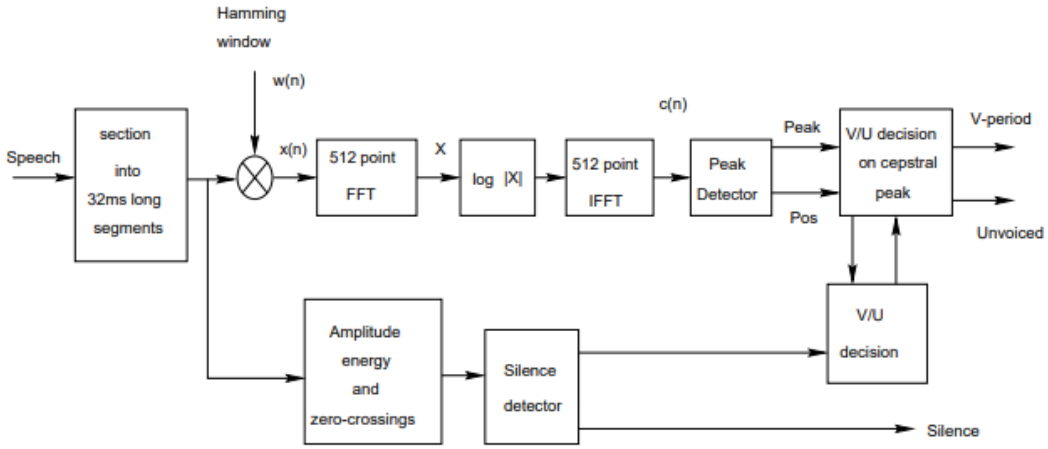


Figure-3: Block diagram of the cepstrum pitch estimator

(iv) When we take IFFT of the signal, we find a slowly varying function of vocal tract cluster near the origin and a rapidly varying function appearing as regular pulses away from the origin.

(v) We can now use a cepstral window allowing the pitch information (the rapidly varying function) to pass through.

(vi) The FFT output of this windowed cepstrum will be spectrum with only a rapidly varying function.

(vii) We tract the peak of this spectrum, we find the pitch frequency.

(viii) The slowly varying function of vocal tract is now isolated and hence the possibility of the first formants overlapping with the pitch frequency removed.

Short Term Autocorrelation Function:

Cross correlation tool from signal processing can be used for finding the similarity among the two sequences and refers to the case of having two different sequences for correlation. Autocorrelation refers to the case of having only one sequence for correlation. In autocorrelation, the interest is in observing how similar the signal characteristics are with respect to time. This is achieved by providing different time lag for the sequence and computing with the given sequence as reference.

The autocorrelation is a very useful tool in case of speech processing. However due to the non-stationary nature of speech, a short term version of the autocorrelation is needed. The autocorrelation of a stationary sequence rxx(k) is given by

$$r_{xx}(k) = \sum_{m=-\infty}^{\infty} x(m).x(m+k)$$

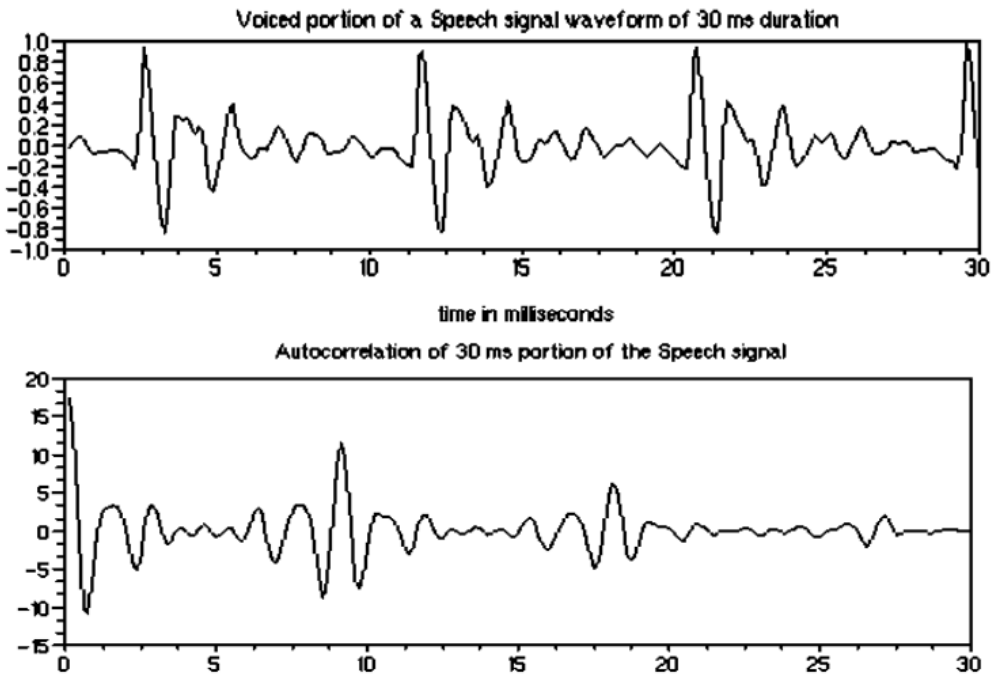
The corresponding short term autocorrelation of a non-stationary sequence s(n) is defined as

$$r_{ss} = \sum_{m=-\infty}^{\infty} s_w(m).s_w(k+m)$$

$$r_{ss}(n,k) = \sum_{m=-\infty}^{\infty} (s(m)w(n-m).s(k+m).w(n-k+m)))$$

Where sw(n)=s(m).w(n-m) is the windowed version of s(n). Thus for a given windowed segment of speech, the short term autocorrelation is a sequence. The nature of short term autocorrelation sequence is primarily different for voiced and unvoiced segments of speech. Hence information from the autocorrelation sequence can be used for discriminating voiced and unvoiced segments.

This Figure shows the segments of voice and the corresponding autocorrelation sequences. The nature of autocorrelation sequence is different for the two cases indicating the difference in case of voiced and unvoiced sequence of speech.



Linear prediction coding (LPC): LPC is one of the good signal analysis methods for linear prediction in speech recognition process. The feature extraction techniques find out the basic parameters of speech. LPC is the most powerful method for determining the basic parameter and computational model of speech. The idea behind LPC is the Speech sample can be approximated as a linear combination of past speech samples. It imitates the human vocal tract and gives robust speech feature. Features that can be deduced from LPC are linear predication cepstral coefficients (LPCC), log area ratio (LAR), reflection coefficients (RC), line spectral frequencies (LSF) and Arcus Sine Coefficients (ARCSIN) . LPC is generally used for speech reconstruction. LPC method is generally applied in musical and electrical firms for creating mobile robots, in telephone firms, tonal analysis of violins and other string musical gadgets.

LPC Algorithm description, strength and weaknesses:

Direct forecast strategy is connected to get the channel coefficients proportionate to the vocal tract by decreasing the cruel square blunder in between the input discourse and evaluated discourse [28]. Straight expectation examination of discourse flag figures any given discourse test at a particular period as a direct weighted conglomeration of going before tests. The linear predictive model of speech creation is given as:

$$\hat{s}(n)=\sum_{k=1}^{p} a_k s(n-k) \quad E3$$

where $\hat{s}$ is the predicted sample, s is the speech sample, p is the predictor coefficients.

The prediction error is given as:

e(n)=s(n)−ŝ (n) E4

Subsequently, each frame of the windowed signal is auto correlated, while the highest autocorrelation value is the order of the linear prediction analysis. This is followed by the LPC analysis, where each frame of the autocorrelations is converted into LPC parameters set which consists of the LPC coefficients . A summary of the procedure for obtaining the LPC is as seen in Figure 2. LPC can be derived by:

am=log[1−km1+km] E5

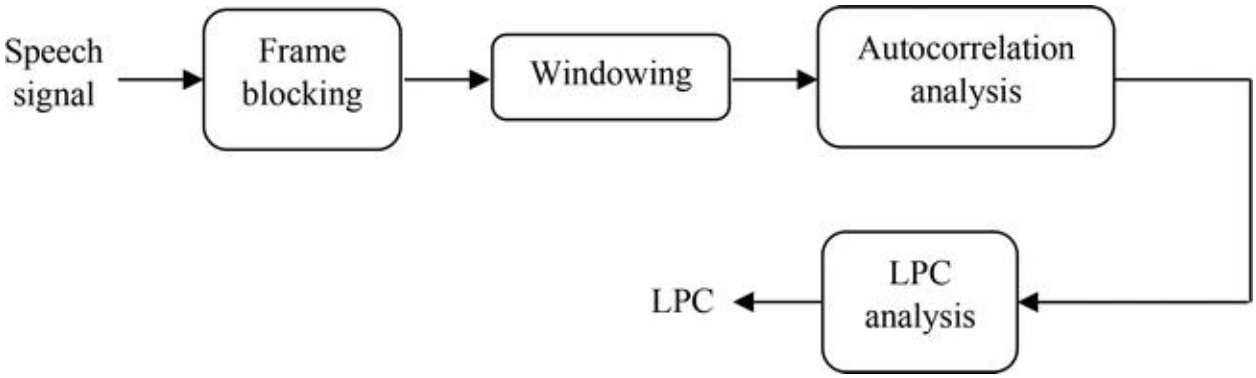


Figure 2.

Block diagram of LPC processor.

where am is the linear prediction coefficient, km is the reflection coefficient.

Straight prescient investigation proficiently chooses the vocal tract data from a given discourse . It is known for the speed of computation and exactness . LPC perfectly speaks to the source behaviors that are consistent and steady . Besides, it is additionally be utilized in speaker recognition framework where the most reason is to extricate the vocal tract properties . It gives exceptionally precise gauges of discourse parameters and is comparatively productive for computation . Conventional direct forecast endures from aliased autocorrelation coefficients . LPC gauges have tall affectability to quantization commotion and might not be well suited for generalization .

Text To Speech Conversion

The text-to-speech (TTS) synthesis is to convert an arbitrary input text into intelligible and natural sounding speech. TTS system includes mainly two parts: natural language processing and digital signal processing. The general block diagram of TTS system is shown in figure 1. Natural language processing contains three steps. They are text analysis, phonetic analysis and prosodic analysis. The text analysis includes segmentation, text normalization, and part of speech (POS) tagger. Phonetic words whereas dictionary based is used for known words. Prosodic analysis is to determine intonation, amplitude and and duration modeling of speech. It describes speaker‘s emotion. conversion is to assign phonetic transcription to each word. There are two approaches in phonetic conversion. They are rule based and dictionary based approaches. Rule based is applied for unknown

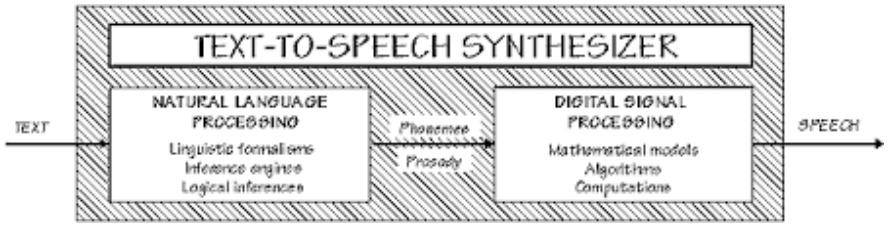


Figure1 .General block diagram of Text to speech (TTS)

Digital Signal Processing refers to speech synthesis. The most important qualities of a speech synthesis system are naturalness and intelligibility. Naturalness expresses how the output sounds like human speech, whereas intelligibility is the easiness with which the output is understood. The technologies for generating synthetic speech waveforms are concatenative synthesis, formant synthesis and articulartorysynthesis . In formant synthesis, speech can be constantly intelligible. It does not have any database of speech samples. So the speech is artificial and robotic. Articulatory synthesis technique is based on the models of human vocal tract for synthesizing speech. Among these, Concatenative synthesis is the primary technology for speech synthesis. It is based on prerecorded natural sounds database. But it is limited to one speaker and usually require more memory capacity. This approach use a real recorded speech as the synthesis units such as: phoneme, syllable, or word and concatenate the units together to produce speech. There are three main sub-types of concatenative synthesis. They are unit selection synthesis, diphone synthesis and domain specific synthesis. In certain systems, this part includes the computation of target prosody (pitch contour, phoneme duration), which is then imposed on the output speech. In this paper, domain specific synthesis is applied to join recorded speech. Text-to-speech synthesis is an useful hardware and software tool in many application areas such as vocal monitoring

system for blind people, web browser, mobile phones, personal computer and so forth. Furthermore, TTS system is currently developed in teaching aids, text reading, and talking books/toys. However, most TTS systems only focus on a limited domain of applications. In this TTS system, three types of speech synthesis such as domain specific synthesis , phoneme based speech synthesis and unit selection synthesis are applied differently depend on the input text. For input numbers and one syllable words , domain specific and phoneme based speech synthesis are applied . For input sentence , unit selection synthesis is used.

METHODOLOGY

Text to speech system has two parts namely natural language processing and speech synthesis (digital signal processing).

Natural Language Processing (NLP)

NLP produces phonetic transcription together with prosodic feature of the input text. In this TTS system, NLP comprises of three main components such as text analysis, phonetic conversion and prosodic phrasing.

Text Analysis

In this TTS system, the input sentence is segmented into token. After tokenization, each word is determined as part o speech (POS) tagging. Part-of-speech is a process assigning correct POS tag to each word in a sentence from a given set of tags. Bigram Model is used for POS tagger. This method is to perform POS Tagging to determine the most likely tag for a word, given the previous and next tags. This can be calculated by using equation (1).For Bigrams, the probability of a sequence is just the product of conditional probabilities of its Bigrams. So if $t1$,2 ,….tn are tag sequence and $w1$,$w2$,.....wn are corresponding word sequence . P(ti |wi) = P(wi |ti) . P(ti |ti+1) (1) Where tidenotes the tag sequence and wi denotes the word sequences. P(wi |ti) is the probability of current word given current tag. Here, P(ti |ti+1) is the probability of a current tag given the previous tag. This provides the transition between the tags and helps capture the context of the sentence.

Phonetic Conversion

In this system, Dictionary based approach is used for phonetic transcription of input word. So, any type of input text that does not include in the dictionary cannot run.

Prosodic Phrasing

Prosodic Phrasing is to assign the phrase of the input text .In this part, chink _n‘ chunk prosodic phrasing, shown in figure 2, is used. In this model, word classes are identified into chink and chunk group .Then, input words are compared with chink or chunk group. Prosodic phrase break

is automatically set when a word belonging the chunks groups. This method basically corresponds to function and content word classed, with some minor modification.

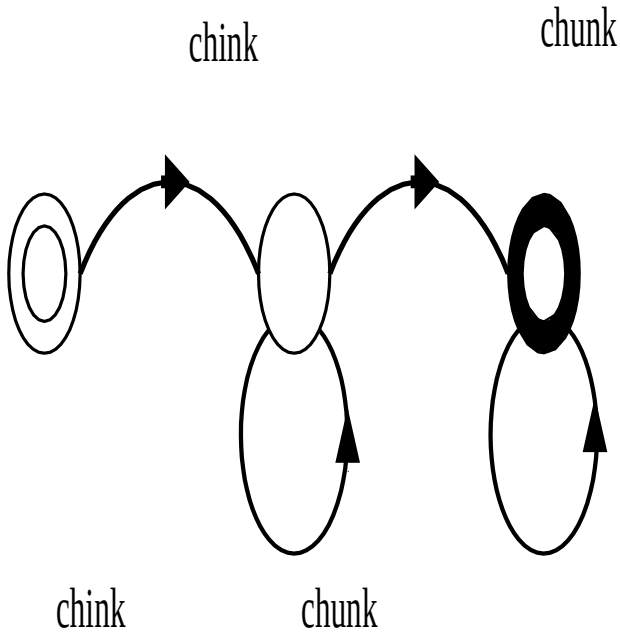


Figure 2. Simple prosodic phrase chink „n chunk

Speech synthesis

The speech synthesis is to produce speech as natural and intelligible sound .There is many methods in speech synthesis. Among these, concatenative speech synthesis is natural in

comparison with other methods. In this TTS system, sub-types of concatenative synthesis such as unit selection speech synthesis, phoneme based speech synthesis and domain specific synthesis are applied.

Unit Selection Speech synthesis

This algorithm selects an optimum set of acoustic units from the speech database to match with the given phoneme stream and target prosody. A selection mechanism using two cost functions – target cost and concatenation (join) cost is applied to find the best sequence of units .The target cost function typically consists of several subcomponents of phonological features such as identity of its context, positional features and numerical features such as phrasing. The target cost, $C\ t\ (ti\ |\ )$ can be computed by the following equation (2) $C\ t\ (ti\ |ui\ )= wj\ q\ t\ j$=1 $Cj\ t\ (tt\ |ui\ )$ (2) where p represents the number of the target cost components,$wj\ t$ and is a feature weight of

the j-th component. The concatenation or joint cost function accounts for the acoustic matching between pairs of consecutive candidate units and it can be calculated by using equation (3) $C\ c$ (ui−1 |ui)= $wj\ q\ c\ j$=1 $Cj\ c$ (ui−1 |ui) (3) where q represents the number of the concatenation cost components The unit selection module is to find the speech unit sequence which is described in equation (4) $C1\ n$=minC($t1\ n$,$u1\ n$) (4) The selection of the optimal speech unit sequence incorporates a Viterbi search.

Phoneme based speech synthesis

Phonemes are the small pieces of speech unit. English language has about 44 phonemes of which 22 sounds are vowels and 22 sounds are consonants. Phoneme based speech synthesis is the concatenation of phonetic units to form word. . Using phonemes as the synthesis unit requires a small storage, but it causes little discontinuity between adjacent units.

Domain-specific Synthesis

Domain-specific synthesis concatenates pre-recorded words and phrases to create complete utterances. The technology is very simple to implement .The level of naturalness of these systems can be very high because the variety of sentence types is limited, and they closely match the prosody and intonation of the original recordings.

IMPLEMENTATION

Different types of speech synthesis systems such as domain specific synthesis, phoneme based synthesis and unit selection synthesis is implemented in this TTS system. 3.1 Domain Specific Synthesis The flowchart of text to speech synthesis based on domain specific synthesis for numbers is illustrated in figure.3. In this part, only numbers are considered as input text.Firstly, speech is recorded for each number and converted to wav file format. Database (lexicon) for digit 0 to 9 is also constructed to compare with the input number .Respective speech are chosen digit by digit based on the numbers. These sounds are then concatenated to generate the wav file. If the input is one digit, the speech can be produced directly. When the input is two or more digits, it is necessary to concatenate each digit.

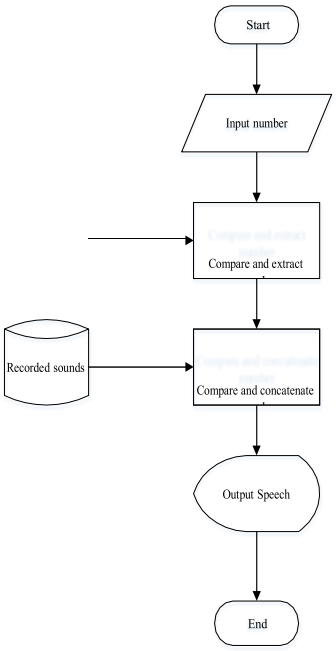


Figure .3 Flowchart of text to speech synthesis based on domain specific synthesis for numbers

Phoneme based speech synthesis

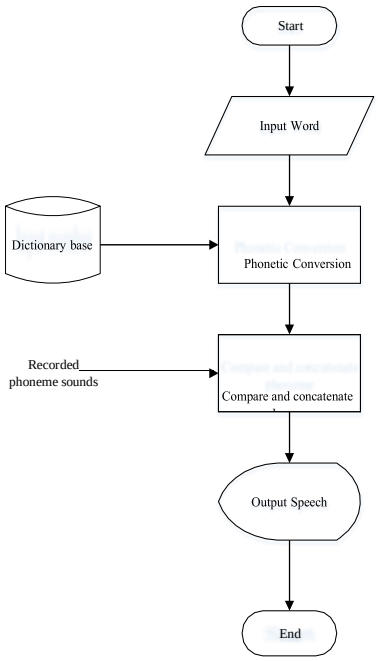


Figure 4. Flowchart of Phoneme based text to speech synthesis for words

The flowchart of phoneme based text to speech synthesis for words is shown in figure .4.In this part; the input text is considered only syllable word to produce speech as natural. Firstly, the input word is given from the keyboard of computer .In the next step, it is necessary to convert from word to phonetic transcription which is also called ―grapheme to phoneme‖ conversion. Dictionary based approach, more exact than rule based approach, is applied in this step. Then, phoneme sounds are concatenated by depending on the phonetic transcriptions of word to produce speech.

Unit Selection Speech Synthesis

In this system, input text of abbreviations, numbers and symbols are not considered. Figure .5 shows the block diagram of TTS using unit selection synthesis. In natural language processing, firstly the input sentence is spitted into words and Part of speech (POS) tagger is assigned to each word by using bigram method. Then, each word is converted to phoneme transcription with the use of dictionary based approach. Simple prosodic phrase chink _n‘

chunk are applied in this system to get speech naturally. These linguistic features are taken as an input of unit selection.

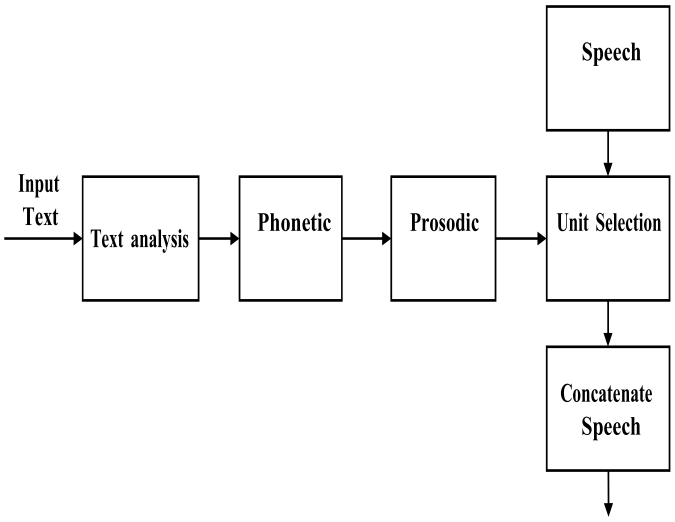


Figure 5. Block diagram of Text to Speech using Unit Selection Synthesis

In speech synthesis, speech is recorded at sampling rate of 16kHz. Then these recorded speeches are segmented phonetically in discrete time domain. So the segmented units are stored according to their sample values. There are two approaches in speech segmentation. They are handlabeling and automatic speech segmentation. In this TTS system, hand-labeling is applied because database is small. But it is time consuming and has little errors in this method. Unit selection algorithm selects the best acoustic units which match the target linguistic features. Then these units are concatenated to produce speech.

Speech enhancement (SE):

Speech enhancement (SE) refers to improve both the speech quality and intelligibility from their corrupted version by applying the different algorithms. The significant objectives of SE can be grouped into the echo cancellation, removal of background or environmental noise, the process of artificially bringing specific frequencies into the speech signal, and reverberation suppression. It is classified into a specialized area of studies such as single-channel and multi-channel SE. It can be categorized into supervised, semi-supervised, and unsupervised SE based on learning.

As a simple medium for data communication, speech is frequently used in daily life. Speech signals are widely used in speech processing systems, such as hearing aids, speech recognition, portable applications, and the like. However, the noise in the environment or the real world will degrade the quality and intelligibility of the speech signal. Therefore, in a single-channel speech enhancement (SE) framework, estimating clean speech signals from noisy speech signals is a difficult and challenging task. Because in some cases, a large part of the noise is non-stationary and may have speech-like characteristics. So, there is always a requirement for the concealment of non-stationary noises. The purpose of the SE algorithm is to improve the quality and intelligibility of speech without significantly degrading it by suppressing interference noise.

Evaluation Metrics:

the subsequent eight indicators are used to measure the performance of SS: Source Distortion Ratio (SDR), Signal-to-Interference Ratio (SIR), Signal-to-Artifacts Ratio (SAR), Perceptual

Evaluation of Speech Quality (PESQ), Short-Time Objective Intelligibility (STOI), Hearing-Aid Speech Perception Index (HASPI), Hearing-Aid Speech Quality Index (HASQI), and average frequency-weighted segmental SNR (fwsegSNR) metrics.

The SDR value approximates the overall speech quality, and it is the proportion of the intensity of the input signal to the intensity of the dissimilarity among input and reformed signals. The higher SDR scores regulate the recovered performance.

$$\mathrm{SDR} = 10\log_{10}\frac{\|\mathbf{x}_{\mathrm{target}}\|_1^2}{\|\mathbf{e}_{\mathrm{interf}}+\mathbf{e}_{\mathrm{noise}}+\mathbf{e}_{\mathrm{artif}}\|_1^2},$$

where $\mathbf{x}_{\mathrm{target}}$, $\mathbf{e}_{\mathrm{interf}}$, $\mathrm{e}_{\mathrm{noise}}$, and $\mathbf{e}_{\mathrm{artif}}$ are the targeted source, the interference error, the perturbation noise and the artifacts error, respectively.

In addition to the SDR, the SIR captures the error caused by failure to remove interfering signal information during the SS procedure. A higher value of SIR relates to higher separation quality.

$$\mathrm{SIR} = 10\log_{10}\frac{\|\mathbf{x}_{\mathrm{target}}\|_1^2}{\|\mathbf{e}_{\mathrm{interf}}\|_1^2}.$$

The signal-to-artifacts ratio determines the energy ratios given in decibels between the estimated desired signal and the distortion, interferences, and artifacts. A higher value of SAR relates to higher separation quality.

$$\mathrm{SAR} = 10\log_{10}\frac{\|\mathbf{x}_{\mathrm{target}}+\mathbf{e}_{\mathrm{interf}}+\mathbf{e}_{\mathrm{noise}}\|_1^2}{\|\mathbf{e}_{\mathrm{artif}}\|_1^2}.$$

The PESQ is nominated for the objective quality assessment and is commonly used to measure speech signals' quality superiority. It deals with scores ranging from -0.50 to 4.50, where the higher scores lead to more outstanding speech quality. PESQ measures the combination of only two parameters – one symmetric disturbance ($\mathrm{d}_{\mathrm{SYM}}$) and one asymmetric disturbance ($\mathrm{d}_{\mathrm{ASYM}}$), provides a good balance between prediction accuracy and the ability to simplify, described in [130].

$$\mathrm{PESQ}_{\mathrm{MOS}} = 4.5 - 0.1\,\mathrm{d}_{\mathrm{SYM}} - 0.0309\mathrm{d}_{\mathrm{ASYM}}.$$

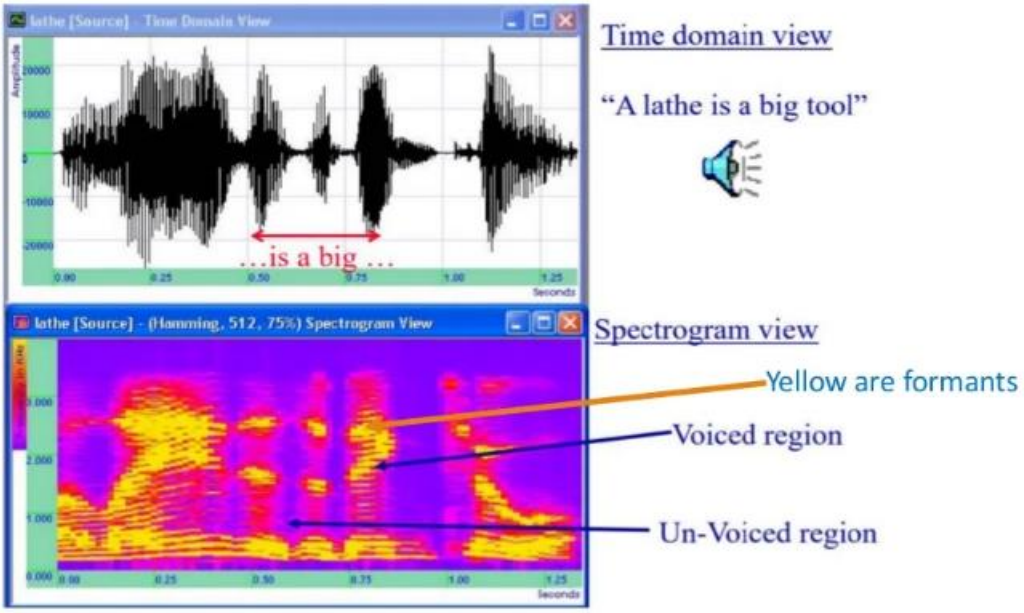
STOI is a state-of-the-art speech intelligibility indicator and deals with the correlation between the short-term temporal envelopes of the clean and separated speech signals. Its value ranges from 0 to 1, with a higher STOI score indicating better intelligibility. We measure STOI, in light of a correlation coefficient among the transient wrappers of the perfect and estimated

speech, in a tiny time frame overlapping fragments. It is a function of the original and corrupted speech, represented by x and $\tilde{x}$, correspondingly.

$$\mathrm{STOI} = \mathrm{Avg}\left(\frac{(\mathbf{x}-\boldsymbol{\mu}_{\mathbf{x}})^{\mathrm{T}}\,(\tilde{\mathbf{x}}-\boldsymbol{\mu}_{\tilde{\mathbf{x}}})}{\|\mathbf{x}-\boldsymbol{\mu}_{\mathbf{x}}\|_1\|\tilde{\mathbf{x}}-\boldsymbol{\mu}_{\tilde{\mathbf{x}}}\|_1}\right).$$

Spectrograms

- The time-varying spectral characteristics of the speech signal can be graphically displayed through the use of a tow-dimensional pattern.
- Vertical axis: frequency, Horizontal axis: time
- The pseudo-color of the pattern is proportional to signal **energy** (red: high energy)
- The resonance frequencies of the vocal tract show up as **"energy bands"**
- Voiced intervals characterized by striated appearance (periodically of the signal)
- Un-Voiced intervals are more solidly filled in



Linear Predictive Analysis: Linear prediction (LP) is one of the most important tools in speech analysis. The philosophy behind linear prediction is that a speech sample can be approximated as a linear combination of past samples. Then, by minimizing the sum of the squared differences between the actual speech samples and the linearly predicted ones over a finite interval, a unique set of predictor coefficients can be determined. LP analysis decomposes the speech into two highly independent components, the vocal tract parameters (LP coefficients) and the glottal excitation (LP residual). It is assumed that speech is produced by exciting a linear time-varying filter (the vocal tract) by random noise for unvoiced speech segments, or a train of pulses for voiced speech.

Figure 4 shows a model of speech production for LP analysis. It consists of a time varying filter H(z) which is excited by either a quasi-periodic or a random noise source.

The most general predictor form in linear prediction is the autoregressive moving average model where the speech sample s(n) is modelled as a linear combination of the past outputs and the present and past inputs. It can be written mathematically as follows

$$s(n) = -\sum_{k=1}^{p} a_k s(n-k) + G\sum_{l=0}^{q} b_l u(n-l), \qquad b_0 = 1 \tag{1}$$

where ak,1 ≤ k ≤ p, bl, 1 ≤ l ≤ q and gain G are the parameters of the filter. Equivalently, in frequency domain, the transfer function of the linear prediction speech model is

$$H(z) = \frac{1 + \sum_{l=1}^{q} b_l z^{-l}}{1 + \sum_{k=1}^{p} a_k z^{-k}}. \tag{2}$$

H(z) is referred to as a pole-zero model. The zeros represent the nasals and the poles represent the resonances (formants) of the vocal-tract. When ak = 0 for 1 ≤ k ≤ p,

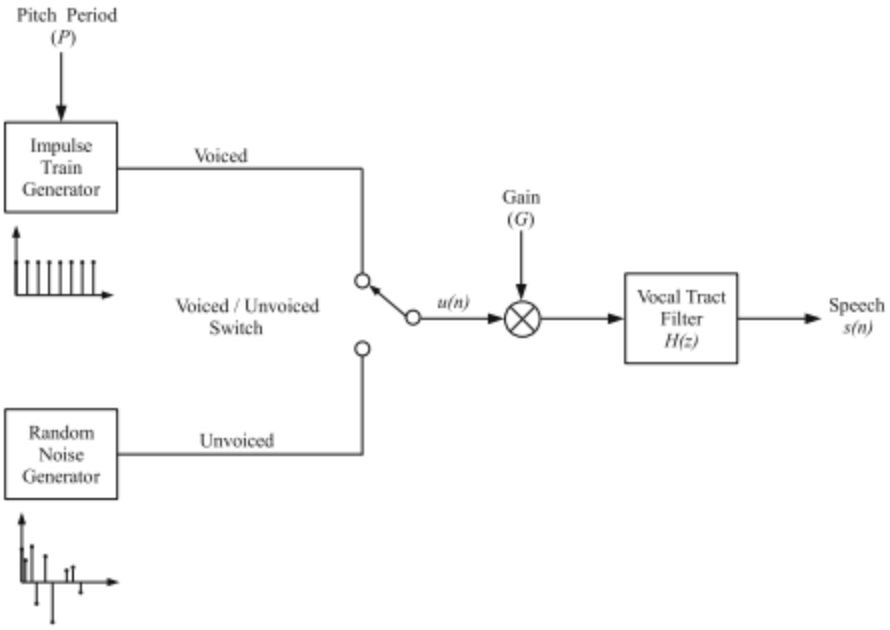


Figure-4: Model of speech production for LP analysis

H(z) becomes an all-zero or moving average (MA) model. Conversely, when bl = 0 for $1 \leq l \leq q$, H(z) becomes an all-pole or autoregressive (AR) model. For non-nasal voiced speech sounds the transfer function of the vocal-tract has no zeros whereas the nasals and unvoiced sounds usually includes the poles (resonances) as well as zeros (anti resonances). Generally, the all-pole model is preferred for most applications because it is computationally more efficient and it's the acoustic tube model for speech production. It can model sounds such as vowels well enough. The zeros arise only in nasals and in unvoiced sounds like fricatives. These zeros are approximately modelled by including more poles. In addition, the location of a poles considerably more important perceptually than the location of a zero. Moreover, it is easy to solve an all-pole model. To solve a pole-zero model, it is necessary to solve a set of nonlinear equations, but in the case of an all-pole model, only a set of linear equations need to be solved. The transfer function of the all-pole model is

$$H(z) = \frac{G}{1 + \sum_{k=1}^{p} a_k z^{-k}}. \quad \textbf{(3)}$$

The number p implies that the past p output samples are being considered, which is also the order of the linear prediction. With this transfer function, we get a difference equation for synthesizing the speech samples s(n) as

$$s(n) = -\sum_{k=1}^{p} a_k s(n-k) + Gu(n) \quad \textbf{(4)}$$

where the coefficients ak's are known as linear predictive coefficients (LPCs) and p is the order of the LP filter. It should be selected such that there are at least a pair of poles per each formant. Generally, the prediction order is chosen using the relation

$$p = 2 \times (BW + 1) \quad \textbf{(5)}$$

where BW is the speech bandwidth in kHz.

Speech Recognition:

Speech Recognition can be defined as the ability of a machine or program to identify words or phrases in spoken language in a machine-readable format. It is the process of converting a speech signal into word sequences by implementing computer algorithms or programming.

Automatic Speech Recognition, often known as ASR, is a technique that converts human speech into text that can be read by using either machine learning or artificial intelligence (AI) technology. Over the course of the last decade, the field has expanded at an exponential rate, and now automatic speech recognition systems can be found in many of the most popular applications that we use on a daily basis, including TikTok and Instagram for real-time captions, Spotify for podcast transcriptions, Zoom for meeting transcriptions, and many more.