Introduction:
After passing through a microphone, our voice is converted into an analogue signal of frequencies about 20 to 20 kHz components. In order to convert this analogue signal into digital form, we have to reduce the bandwidth to a finite value. It has been seen that most of the information in the human voice signal lies between 300 to 3400 kHz. This bandwidth is achieved by passing the voice analogue signal through a BPF of the mentioned bandwidth. Then the signal is converted to a digital bit stream using various techniques depending on the type of the transmission system used. For efficient use of the limited spectrum in use in digital cellular systems, several techniques are adopted i.e. cell’s small geographical size, limited signal power, ad so on, to minimize the signal bandwidth [3]. For a given cell size and signal bandwidth, if the bit rate representing the voice signal is reduced, more users can be multiplexed to a radio channel. In PCM, a bit rate of 64kbits/s is achieved by sampling the voice signal at a rate of 8000 samples/s and 8 binary bits representing a sample. This bit rate is further reduced to 32kbps by using differential PCM and down to 16kbps by Delta modulation [1]. But in the case of cellular mobile radio, the allotted radio band is really limited as compare to the number of service users e.g. in GSM, a band of 25 MHz is allowed for uplink and same size for downlink communications while the number of service users are in hundreds of millions. So the bit rate above 16kbps is still not acceptable for digital cellular systems. There are some systems, like LPC, developed to attain a bit rate much lower than 16kbps on the cost of low speech quality and system complexity. In order to achieve a best trade off between speech quality and bit rate, GSM uses a coding technique called regular Pulse Excitation/Linear Predictive Coding (RPE/LPC) and a final full data rate of 13kbps is achieved [1]. This coding is based on a technique called analysis by synthesis predictive coding in which instead of transmitting the original speech signal, just the information about speech production process is transmitted and at the receiver end, the original voice wave form is produced using this information. GSM makes use of the fact that during a telephone conversation one speaker only speaks for about 40% of the total conversational time and for rest of the time he is either listening or breathing and the voice signal is only transmitted when he is talking [2]. In this report, we are going to discuss all of the major techniques and the steps used in GSM for speech digitisation. The discussion will be a comprehensive summery of the
operational and visual point of view rather than the theoretical details because of the limitation on the length of the report.

**Analogue to Digital Conversion**

After passing through microphone, voice analogue signal is converted into digital form. Signal waveform is sampled at a rate of 8ksamples/s and each sample is represented by 13 binary digits. This gives a bit rate of 104kbps. In order to reduce this huge data rate to 13kbps, data is compressed by passing it through a speech encoder RPE/LPC based on principle of Analysis by Synthesis Speech Coding as mentioned above [3].

**Analysis by Synthesis Speech Coding**

In this technique, the excitation source e.g. lungs is modelled by regularly spaced pulses and their amplitudes are calculated using feedback loop. Then a synthesised waveform is produced by a synthesis filter using these pulses as shown in the fig. below.

![General Model for Analysis by Synthesis Encoder](image)

**Fig. General Model for Analysis by Synthesis Encoder.**

The coefficients of the filter are adjusted in such a way to minimise the difference between actual and synthesised waveform. Then the output in the form of pulse amplitude and the coefficients of the filter are transmitted to the receiver. The decoder at the receiver uses same principle to reproduce a synthesised speech by using this information [2].

**Regular Pulse Excitation/Linear Predictive Coding**

After a comparison between SBC/APCM, SBC/ADPCM, MPE/LTP and RPE/LPC based on speech quality, data rate, processing delay, bit error rate, achievable mean opinion score(MOS) and computational complexity, it had been decided by the
standardisation authority that RPE/LPC best suits GSM. In RPE/LPC, data rate is further compressed by adding a LTP loop in the process. A detailed figure below shows the process. In pre-processing, a one pole filter of transfer function: 

\[ H(z) = 1 - C_1 \cdot z^{-1} \]

is to emphasis the high frequency and low power part of the speech spectrum.

![Fig.2. RPE/LPC (Source: [1])](image)

Then each block of 160 samples is passed through a hamming window which reduces the oscillations at the edges of the block without disturbing it middle part but it does not change the power of the speech [1].

\[ S_{pws}(n) = S_{ps}(n) \cdot C_2 \cdot (0.54 - 0.46 \cos(\frac{2\pi n}{l})) \]

Where Spsw is the speech segment after Hamming window and Sps is the one before it. Practically the value of C1 is taken 0.9 and C2 is 1.5863 [1].

Then a Short Term Prediction analysis filter computes 9 autocorrelation coefficients by the equation:

\[ R(k) = \sum_{n=1}^{L-k} S_{pws}(k) S_{pws}(n-1) \]

Where k=0, 1, 2, . . . 8.

Eight reflection coefficients ki are measured on the basis of there autocorrelation coefficients then these reflection coefficients are translated into logarithmic area ratios LAR(i) using the following formula.

\[ LAR(i) = \log_{10} \left( \frac{1 + k(i)}{1 - k(i)} \right) \]
Dynamic range and PDF of $\text{LAR}(i)$ are shown in the graph below:

Fig.3. (Source: [1])

All LAR$(i)$ are quantized for the values of $|k(i)|$ in the range from 0 to 1 in order to simplify the real-time implementation as given below:

$$\text{LAR}'(i) = \begin{cases} 
    k(i) & \text{for } |k(i)| < 0.675 \\
    \text{sign}[k(i)][2k(i)] & \text{for } 0.675 < |k(i)| < 0.95 \\
    \text{sign}[k(i)][8k(i)] & \text{for } 0.95 < |k(i)| < 1 
\end{cases}$$

These LAR$(i)$ are decoded at the local transmitter end denoted by LAR$'//(i)$ and at also transmitted at the speech decoder at the receiver end [1]. LAR parameters are then interpolated to LAR$'//(i)$ in order to create a smoothness at the frame edges around STP. Then locally decoded reflection coefficients are determined by converting interpolated LAR parameters back to $k'(i)$. Eventually, at this stage, $k'(i)$ is used to calculate residual $\text{rstp}(n)$ [1].

The speech signal is then processed through the next stage called long term prediction (LTP) analysis which finds pitch period $p_n$ and the gain factor $g_n$ at the minimum value of LTP residual $r_n$ [4]. The LTP delay $D$, maximizing the cross-correlation between the current STP residual $\text{rstp}(n)$ and its previous received and buffered history, minimizes the LTP prediction error. All of 160 samples of STP residual are divided into 4 sub-segments each with length of 40 samples. One STP calculated by finding the cross-correlation between the sub-segment under process and a 40 segment long sample from the previously received 128 sample long STP residual segment. This correlation is minimum at delay $D$ and the current data segment is most similar to its history. If all of the highly correlated data segments are subtracted from the STP, it means most of the redundant data is removed from the information data [1]. LTP filter parameters’ operation is best explained by [1] as below, I do not want to further summarise it: “Once the LTP parameters $G$ and $D$ have been found, they are quantised to give $G'$ and $D'$, where $G$ is quantised only by two bits, while to quantise $D$ seven bits are sufficient. Quantised LTP parameters $(G', D')$ are locally decoded into pair $(G'', D'')$ so as to produce the locally decoded
STP residual \( r'_{LTP} (n) \) for use in the forth coming sub-segments to provide the previous history of the STP residual for the search buffer. Since \( D \) is the integer, we have \( D=D'=D'' \). With the LTP parameters just computed the LTP residual \( r_{STP} (n) \) is calculated as the difference of the STP residual \( r_{STP} (n) \) and its estimate \( r''_{STP} (n) \), which has been computed by the help of the locally decoded LTP parameter \( (G'',D) \) as shown below:

\[
 r_{LTP} (n) = r_{STP} (n) - r''_{STP} (n) \\
 r''_{STP} (n) = G'' r'_{STP} (n-D).
\]

Here \( r'_{STP} (n-D) \) represents an already known segment of the past history of \( r'_{STP} (n) \), stored in the search buffer. Finally the content of the buffer is updated by using the locally decoded LTP residual \( r'_{LTP} (n) \) and the estimated STP residual \( r''_{STP} (n) \) to form \( r'_{STP} (n) \), as shown below:

\[
 r'_{STP} (n) = r'_{LTP} (n) + r''_{STP} (n). \quad [1]
\]

The final step in the speech digitization in GSM LPC is to process the outcome from LTP to determine regular pulse excitation measurements. The residual \( r_{LTP} (n) \) is passed through a band limiting low pass filter of cut off frequency of 1.33 kHz. The smoothed LTP residual \( r_{SLTP} (n) \) is braked down into three excitation candidates and the one with the highest energy level is nominated for the LTP residual. Then first the pulses are normalised with the maximum amplitude in the series of the 13 samples and then each pulse is quantised with three bits quantiser while the log of the block maximum is quantised with six bits [1].

Main job of LPC in GSM coding is to save every 160 samples in a short-term memory and compress the data bits to a small fraction of the total bits. It takes all the excessive redundancies in the speech signal. As one sample is composed of 13 binary bits, total number of bits in 160 samples is 2080. As each sample is taken after 125usec, total time of 2080 bits is 20msec. LPC basically identifies the repetition of data within each 20msec block and assigns pointers to that data. It assigns 9 coefficients which measure the excitation generator’s process and synthetic filtration. The characteristics of the speech i.e. loudness and pitch are measured by the excitation generator and LPC removes all of the redundancies in the speech by generating correlation between the actual speech characteristics and its redundancies. This signal is then passed through LTP analysis function which
compares the data with the data sequences it has stored in its memory earlier. After comparison it selects the most resembled sequence of data from the memory and measures the difference between the received data and the selected data sequence. Then it translates the difference and assigns a pointer which refers to the data sequence selected for comparison. Instead of transmitting the excessive data, only the information that this data is redundant is transmitted to the receiver and the receiver reproduces that data itself. This way LPC compresses the data block of 20msec from 2080bits to 260 bits. 260bits/20msec give a data rate of 13kbps which is full rate speech coding [1].

In accordance with their importance and functionality, this block of 260 bits is classified into three categories: class1a bits, class1b bits and class 11 bits. Class 1a bits consists of filter coefficients, speech sub blocks amplitudes and LPC parameters. This is the most important class of the data block. Second important class is 1b which consists of 132bits. It contains the information about the excitation sequence and long-term prediction parameters. The third important (less important) class of the data is class 11. A amount of 196 bits are added to class 1a and 1b data bits as channel coding in order to detect and remove transmission errors [2].

The figure given below shows a summery of all the steps involve in the process of speech digitization and some channel coding in GSM mobile radio.

![Fig.5. (Source: [3])](image)

**Computer Model of the System using Matlab and Result Analysis**

A computer model of all the processes involved in speech digitisation in GSM given in figure 3 above is described here and their simulation results analysis is provided step by step.
In the following Matlab code, an audio file is taken from PC and plotted.

```matlab
clear all;
close all;
[source FS NBIT]=wavread('sound3_Oct2011.wav');
figure();
plot(source);
```

The output of the audio signal before the BPF is shown below:

![Audio Signal before BPF](image)

---

This speech wave contains frequencies from 20 to 20kHz. This signal is then passed through a butter worth band pass filter of bandwidth 300 to 3400kHz. The code for this filter is as below:

```matlab
LOW=300;
HIGH=3400;
WN=[LOW/(FS/2) HIGH/(FS/2)];
[B, A]=butter(1,WN);
source_bpf=filter(B,A,source);
figure();
plot(source_bpf)
```

This code produces an output as follow:

![Filtered Audio Signal](image)

---

![Fig.6. Audio Signal before BPF.](image)
Here the out band frequencies are removed.
This signal is then passed through ADC. A simulation model up to ADC is as follow:

![Fig.8]

The simulation gives a result as follow:

![The simulation gives a result as follow:]

This simulation encodes the speech signal by in 8192 quantization levels each level is represented by 13 bits.
The next process is LPC. As this process is completed at the receiver end when the speech is synthesised by using the received information, at the moment it is difficult to model all the processes involved in channel coding, transmission and receiving of our speech signal. Let us process our speech signal in analysing and synthesising altogether. A simulation model given below does this job efficiently.

![LPC Analysis and Synthesis of Speech][source: [5]]

![LPC Analysis and Synthesis of Speech](source: [5])

We initiate this process using following code [5]:

```plaintext
frameSize = 160;
```
fftLen = 256;

Here you create a System object to read from an audio file and determine the file's audio sampling rate [5].

```matlab
hfileIn = dsp.AudioFileReader('speech_dft.wav', ... 'SamplesPerFrame', frameSize, ... 'PlayCount', Inf, ... 'OutputDataType', 'double');
```

```matlab
fileInfo = info(hfileIn);
Fs = fileInfo.SampleRate;
```

FIR digital filter is created using following code and used for pre-emphasis [5].

```matlab
hpreemphasis = dsp.DigitalFilter(... 'TransferFunction', 'FIR (all zeros)', ... 'Numerator', [1 -0.95]);
```

Code for Buffer System object is as follow. Its output is of twice the length of the frame size with an overlap length of frame size [5].

```matlab
hbuf = dsp.Buffer(2*frameSize, frameSize);
```

Hamming Window is created by the following code.

```matlab
hwindow = dsp.Window;
```

Autocorrelation is created using following code and it computes the lags in the range [0:12] scaled by the length of input.

```matlab
hacf = dsp.Autocorrelator( ... 'MaximumLagSource', 'Property', ... 'MaximumLag', 12, ... 'Scaling', 'Biased');
```

The following code creates a system object which computes the reflection coefficients from auto-correlation function using the Levinson-Durbin recursion. You configure it to output both polynomial coefficients and reflection coefficients. The polynomial coefficients are used to compute and plot the LPC spectrum.

```matlab
hlevinson = dsp.LevinsonSolver( ... 'AOutputPort', true, ... 'KOutputPort', true);
```

The following code creates an FIR digital filter system object used for analysis and it also create two all-pole digital filter system objects used for synthesis and de-emphasis.

```matlab
hanalysis = dsp.DigitalFilter(... 'TransferFunction', 'FIR (all zeros)', ... 'CoefficientsSource', 'Input port', ... 'Structure', 'Lattice MA');
```

```matlab
hsynthesis = dsp.DigitalFilter( ...
We create a loop using following code which stops when it reaches the end of the input file, which is detected by the AudioFileReader system object.

```matlab
while ~isDone(hfileIn)
    sig = step(hfileIn);
    sigpreem = step(hpreemphasis, sig);
    sigwin  = step(hwindow, step(hbuf, sigpreem) );

    sigacf = step(hacf, sigwin);
    [sigA, sigK] = step(hlevinson, sigacf);
    siglpc = step(hanalysis, sigpreem, sigK);

    sigsyn = step(hsynthesis, siglpc, sigK);
    sigout = step(hdeemphasis, sigsyn);

    step(haudioOut, sigout);
end
```

The analysis portion is found in the transmitter section of the system. Reflection coefficients and the residual signal are extracted from the original speech signal and then transmitted over a channel. The synthesis portion, which is found in the receiver section of the system, reconstructs the original signal using the reflection coefficients and the residual signal.

The input signal is like this:

![Fig.11. Input speech wave](image1)

The coefficients produced by analysis section look like this:

![Fig.11. Coefficients produced by analysis section](image2)
Fig. 12.
At the receiver end the synthesised speech, given below, is almost the same as the original one.

![Graph](image)

Fig. 13.
In this simulation, the speech signal is divided into frames of size 20 ms (160 samples), with an overlap of 10 ms (80 samples). Each frame is windowed using a Hamming window. Eleventh-order autocorrelation coefficients are found, and then the reflection coefficients are calculated from the autocorrelation coefficients using the Levinson-Durbin algorithm. The original speech signal is passed through an analysis filter, which is an all-zero filter with coefficients as the reflection coefficients obtained above. The output of the filter is the residual signal. This residual signal is passed through a synthesis filter which is the inverse of the analysis filter. The output of the synthesis filter is almost the original signal [5].

**References**

[1]: Digital Mobile
[2]: Digital Mobile Radio and TETRA System
[3]: gsmfordummies.com
[4]: Wireless Communications, fourth edition.
[5]: Matlab Help Window